A Case-Based Reasoning System for Knowledge and Experience Reuse. Mobyen Uddin Ahmed, Erik Olsson, Peter Funk, Ning Xiong, in the proceedings of the 24th annual workshop of the Swedish Artificial Intelligence Society, p 70-80, Borås, Sweden, Editor(s):Löfström et al., May, 2007

A Case-Based Reasoning System for Knowledge and Experience Reuse

Mobyen Ahmed, Erik Olsson, Ning Xiong, Peter Funk

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

Abstract. Experience is one of the most valuable assets technicians and engineer have which may have been collected during many years both from successful solutions as well as from very costly mistakes. Unfortunately industry rarely uses a systematic approach for experience reuse. This may be caused by the lack of efficient tools facilitating experience distribution and reuse. We propose a case-based approach along with a domain specific ontology and tool to facilitate experience reuse more systematically in industry. It is important that such a tool allows the technicians to give the problem case in a flexible way to increase acceptance and use. The proposed tool enables more structured handling of experience and can be adapted to different situations and problems. The user is able to input his experience as free text format in a structured and/or semi-structured way. A domain specific ontology combined with vector space model able the system to identify and retrieve relevant similar experiences to reuse.

1. Introduction

The importance of experience sharing is increasing and the potential for increased efficiency, creativity and cross fertilization of experience and ideas is large. The benefit of experience sharing and reuse of a previous knowledge 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 [4][5], diagnostics and condition based maintenance system [6] and health monitoring system [7]. 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 [18].

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 the most crucial step and has the largest impact on economy. Different circumstances may need different solutions for a problem. A standard solution, i.e. 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 a time critical customer order. Identifying past experience relevant to the situation will help the engineers to take a better end more informed decision avoiding previously sometimes very costly mistakes. The need of a domain dependent experience sharing system where experience can be gathered, stored and reused is obvious. This kind of experience sharing could be done within a company, amongst collaborating companies or even with competitive companies if a suitable knowledge and experience trading scheme can be implemented to enable sharing the economical gain from reusing experience. This would increase productivity and efficiency of the area without preventing the spreading of valuable knowledge. For example if the monitoring system notifies a deviation in a machine [6], 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 corrects 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. In this paper we proposed a Case-Based experience sharing solution that may enable reuse of experience in a more efficient way compared to what common practiced 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 [8].

Case-based Reasoning (CBR) is a methodology for problem solving reusing previous experience [2] and also for collecting new experience since every new problem case, once solved becomes a new case that may be stored and reused. The *tf-idf* (term frequency–inverse document frequency) [10] weighting scheme is used in the vector space model [16] together with *cosine* similarity [19] [20] to determine the similarity between two textual cases [9]. Additional domain information often improves the results, e.g. a list of words and their synonyms or dictionaries that provide comparable words [12] [17] and relationships within the words using class and subclass [13]. Our proposed system uses domain specific ontology that represent specific knowledge [14] [15] i.e. relation between the words. For example, in the domain of industrial robotics, an ontology with entities and relationships could be found 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 the words in the ontology can be used in the cases.

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

2. Case-based reasoning for experience reuse

Human solve problems by using both his own experience as well as that learned from other experienced people, simulations, modeling etc. It is always valuable with a second option and providing a system able to provide this by identifying similar and relevant past cases in a tool that technicians would appreciate [18]. The methodology of Case-based reasoning is used to solve new problems often by using existing experience that is obtained by remembering a previous similar situation.

CBR [1, 3] 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 [1].

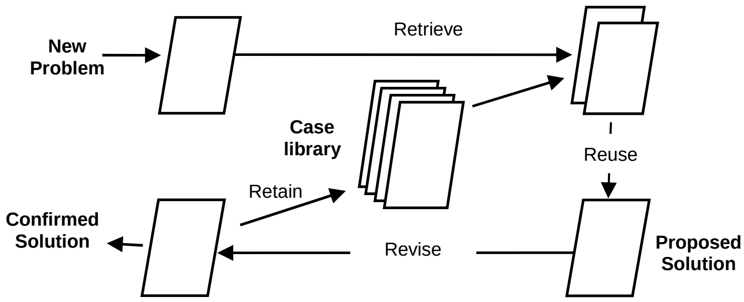

Fig. 1. The case-based reasoning cycle proposed by Amodt and Plaza [1]

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 neighbor search is used to identify similar cases. If it finds any suitable case that is closer to a current problem then the solution is *reused* (after some revision and adaptation 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. Representation of experiences as cases

An experience can be represented as a contextualized piece of knowledge in a case. 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). Finding suitable features and structure of a case for a specific domain is an important issue. If the important features are not extracted and if the structure is too ambiguous the result of the matching will not be good enough. In the mechanical engineering domain, knowledge is often stored and described with free or semi-structured text instead of some predefined structure. Moreover often this knowledge is defined using different categories and different names of labels. In our system cases are represented as a free text but in a structured way. A sample screen shot for the case structure is shown in fig 2.

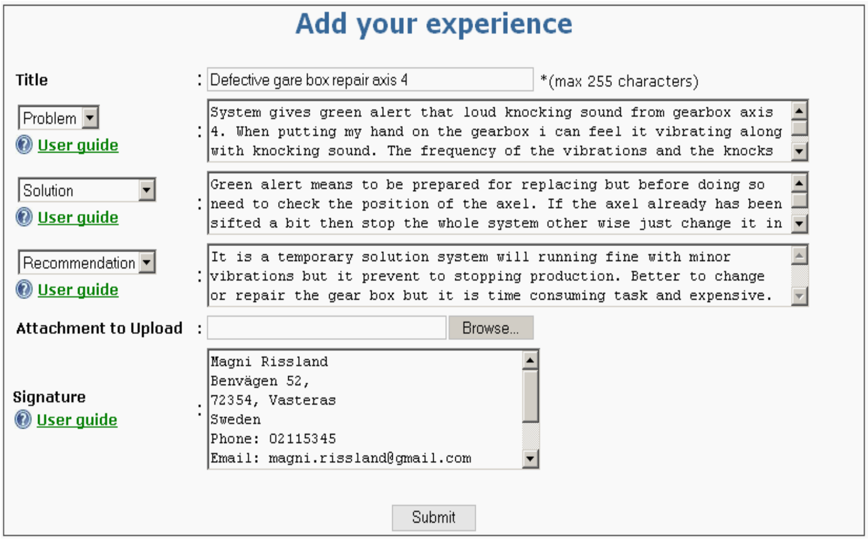

Fig. 2. General case structure and its content

The interface of adding an experience can adapt to different situations and problems because the structure of an experience is flexible and user can define their own experience structure by themselves. For example someone can define experience with problem, solution and recommendation; other can define with question, answer and reflection. In figure 2 users can select different criteria to represent his/her structure of the experience, enter their experiences in free text, upload files as attachment and finally enter his/her contact address and all these fields are optional here and the user do not need to login the system. Thus the system has the facility to use it in a flexible and easy way to increase the user acceptance.

4. Retrieval of similar cases

A CBR system generally includes 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 have potential to be reused. The procedure of case retrieval begins with identifying the most important features, then doing some search and matches, and ends up with selecting the most similar case(s). The different steps in the retrieval of similar case(s) in the system are shown in fig 3.

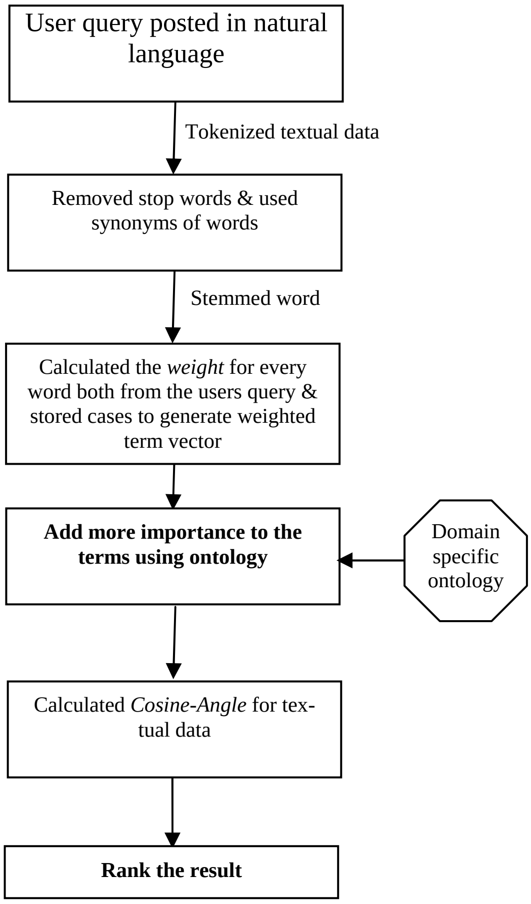

Fig. 3. Different steps for case retrieval.

User's specific problem description is given to the system through the user interface in a natural language format. The text tokenizer algorithm decomposes the whole textual information into sentences, and then into individual words. A filtering step is needed to improve retrieval effectiveness because of the huge amount of words. The following three steps are used:

1. Remove the stop-words and special characters by the stop-words and special characters blacklist both from the users' query and stored cases.
2. A list of synonyms of the words are used to reduce the number of terms and Porter stemming algorithm [17] helps stemming the words that provide the ways of finding morphological variants of search term. After calculating the weight for each word, these words are represented as terms in a vector space.

3. Improve the importance assessments for candidate terms before measuring the cosine similarity value for the textual information between the stored case and user's query case by using domain specific ontology.

4.1. Weighting the terms ($W_{i,j}$)

The weighting terms method [11] is chosen from the different algorithms (such as $W_{i,j}$ method, Relating term precision to term frequency method, Term discrimination method, etc) to calculate the weight of each term from the stored cases and the inputted user's query to perform further matching. The general equation for $W_{i,j}$ can be shown by equation (1).

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

Where, $W_{i,j}$ is the weight of term T_j in the case C_i, $tf_{i,j}$ is the 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 df_i is the number of cases where term T_j occurs at least once.

According to the vector space model the new case is processed and stored in a separate table in the database. First, an index of the terms from the case collection is constructed and the frequency of the terms ($tf_{i,j}$) appearing in each case (C_i) and new query case (Q) is counted. Then, the case frequency (df_i) from the collection of cases and the inverse case frequency (idf_j) are calculated and finally, the $tf_{i,j}* idf_j$ product gives the weight for each term.

4.2. Enhanced term vector using ontology

Each word of a case can be treated as a term and it is easy to calculate 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 or terms have a complex relationship (for example, the term axel and gearbox), those can be defined by ontology and the weight of those terms can be increased for that case by ontology. We can enhance the weight to the vector terms for each case as below:

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 exist in the case then the term T_o can be added as a *new* term by the same importance as the weight of the source term i.e. 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 exists in the case then the strength of relationship between the term T_f and T_o can be added to the original weight (i.e. score of *tf-idf*) of those terms.

3. If more than one term in a case are related to a term T_o in the ontology then those terms of that case will get more importance by adding their relationship strength to their original weight (i.e. score of *tf-idf*).
4. If a term T_f in a case is related to more than one term in the ontology then the normalized strength of their relationship can be added to the original weight of source term T_f.

An example is shown in fig.4 on how the ontology helps to improve the weight vector.

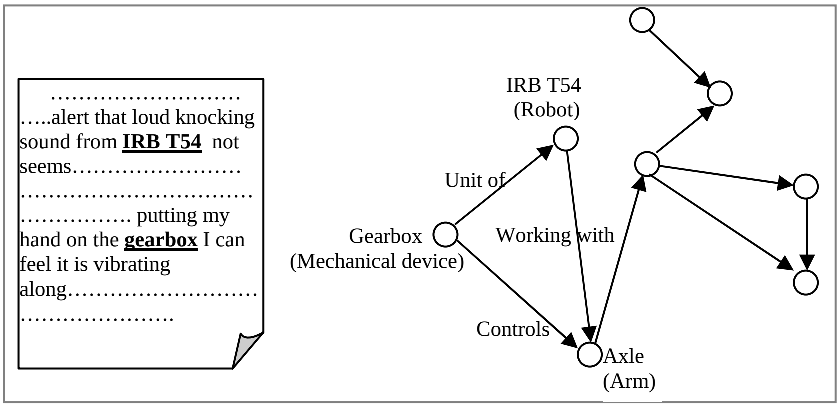

Fig. 4. Weighting term vector using ontology

From figure 4 "gearbox" is a term that appears both in the case text and in ontology, has a relation with another term "axle" in the ontology but the term "axle" does not exist in the case text, so the term "axle" is important for this case and can be added according to condition 1.

Again the terms "IRB T54" and "gearbox" both are already existing in the case text and as well as has a relation in the ontology so the value of their strength of relationship for those two terms ("IRB T54" and "gearbox") will increase their importance (condition 2).

Terms "IRB T54" and "gearbox" are related with another term "axle" in the ontology so the term "axle" will get more importance according to condition 3. Condition 4 is the vice versa of the condition 3.

Thus the terms will get importance assessments depending on ontology and they allow us to make a more efficient calculation in the similarity matching.

4.3. Similarity functions

To find the similarity between a stored case vector C_i and a new case query vector Q we implemented cosine similarity function [11] [19] [20] for the textual information. This ratio is defined as the cosine of the angle between the vectors, with values between 0 and 1 and can be calculated by the equation 2.

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

Where $Cos\theta_{C_i}$ is the *cosine of the angle* between a stored case and a query case which is defined as the similarity function *Sim (Q, C_i)*. For a stored case dot product is calculated with the query case by *Q.C_i* where zero products ignored, next vector lengths $\|Q\|\|C_i\|$ are calculated for a stored case and a query case (zero terms also ignored) where $w_{i,j}$ and $w_{q,j}$ are weights calculated through equation 1.

5. Results

The system formulates a ranking of stored cases based on the angle cosine value for each case paired with the query description. All the cases are listed according to the percentage where 100% means a perfect match on all the relevant features extracted from the text. The problem, solution and its related parts can be presented. A screen shot for such a search is shown in fig 5.

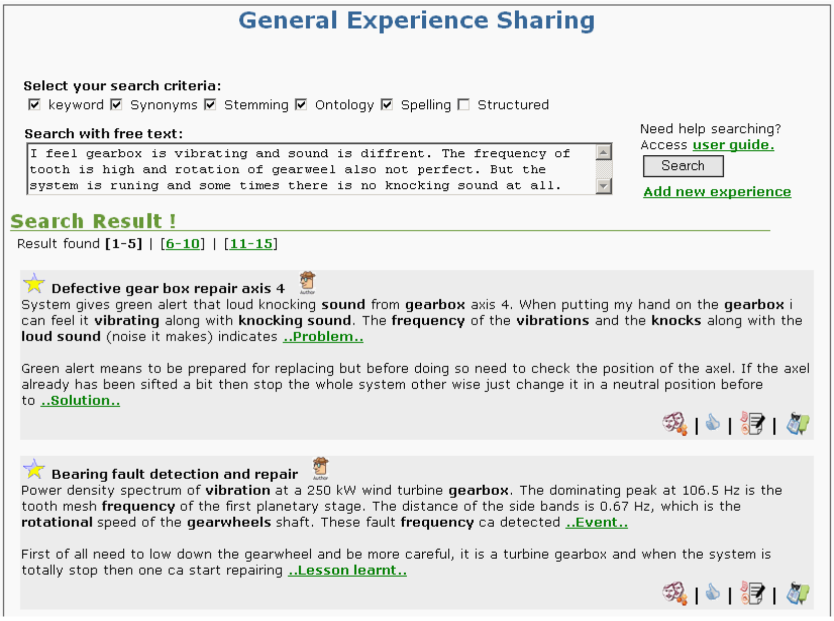

Fig. 5. User interface for experience search

The user interface in figure 5 is for search of experience consisting of a text area along with check boxes where user can enter textual query and can choose different searching criteria. After finishing the search, depending on the search criteria the most relevant case(s) are ranked with case title and author's information along with the score (similarity value). The tooltip of blue-yellow star presents the similarity value and tooltip of author's symbol presents details information about the author. Four symbols are presented below each experience, the first one facilitates the user to make further matching that is it allows to show the same type of experiences, user can make rate on each case by the second symbol, user can report on the experience according to reviewing the experience which can be done by the third symbol and finally with the last symbol user can see other comments on the selected experience and at the same time user can give his/her own comments.

User can use several search criteria, for example, only keywords matching which makes the search depend only on keywords that is exact word matching between query and stored cases. User can get similar types of searching by choosing synonyms and stemming but by selecting ontology user can get more accurate matching results. Right now matching is made within the whole text by merging the problem, solution and recommendation text fields but we could further extend to use the structured searching where problem description can have more importance than the solution and recommendation. System can also detect spelling error through the dictionary where WordNet is used. By default all searching criteria is pre-selected and user has the option to change and get a different matching result.

In the reuse step the retrieved experiences are reused to solve or share with a new experience. In our system user can give feedback and comments about the experience(s); for example user will rate for a retrieved experience on how much the experience has been matched with his/her current experience and how valuable the experience is. This information will help the system for evaluating our proposed matching algorithm.

6. Summary and conclusions

Human experience is a valuable asset and could be more valuable if those can be manipulated and reused in an efficient way. We have shown how case-based reasoning is useful for making tools for experience reuse and sharing; such kind of experience sharing tool is also able to reduce costs in an industrial environment by transferring relevant knowledge to an engineer to solve a problem. From this tool it is not only identifies valuable experience for the current situation, it also enables avoidance of expensive mistakes. Since cases also have case author's information 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 textual CBR approach where domain specific ontology improves the similarity accuracy. Also transferring experience through cases is a form that is liked by technicians and they recognize such a system as a useful tool giving them decision support as a second opinion. This acceptance is only on a theoretical level until now and once the system is ready for a field tests the actual need and requirements of the users' can be collected leading to further improvement of the approach.

References

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

2. Watson, "Case-based reasoning is a methodology not a technology", AI-CBR, University of Salford, Salford M5 4WT, UK ,Received 1 December 1998; accepted 17 March 1999

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

4. Delaître, S. and Moisan, S., 2000, "Knowledge Management by Reusing Experience", 12th International Conference on Knowledge Engineering and Knowledge Management, pp. 304-311

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

6. 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.

7. 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.

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

9. Rosina O. Weber, Kevin D. Ashley and Stefanie Brüninghaus. "Textual case-based reasoning". The Knowledge Engineering Review, Vol. 00:0, 1–00. © _ 2005, Cambridge University Press DOI: 10.1017/S0000000000000000 Printed in the United Kingdom.

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

11. Dr. E. Garcia, "Tutorial on Cosine Similarity Measures". http://www.miislita.com/information-retrieval-tutorial/cosine-similarity-tutorial.html, last referred 20-03-2006.

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

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

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

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

16. 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.

17. 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.

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

19. Doireann C, Joe C, Anne D, John D, and John S, 2003, “The Use of Data Mining in the Design and Implementation of an Incident Report Retrieval System", Proceedings of the Systems and Information Engineering Design Symposium.

20. Gupta, K.M. Montezemi, A.R., 1997, “Empirical Evaluation of Retrieval in Case-Based Reasoning Systems Using Modified Cosine Matching Function”, IEEE transactions on systems, man, and cybernetics—part a: systems and humans, vol. 27, no. 5.