

Document downloaded from:

<http://hdl.handle.net/10251/78887>

This paper must be cited as:

Heras Barberá, SM.; Garcia Pardo Gimenez De Los Galanes, JA.; Ramos-Garijo Font De Mora, R.; Palomares Chust, A.; Julian Inglada, VJ.; Rebollo Pedruelo, M.; Botti V. (2006). CBR model for the intelligent management of customer support centers. En Lecture Notes in Computer Science. Springer Verlag (Germany). 663-670. doi:10.1007/11875581_80.



The final publication is available at

https://link.springer.com/chapter/10.1007/11875581_80

Copyright Springer Verlag (Germany)

Additional Information

CBR Model for the Intelligent Management of Customer Support Centers

Stella Heras Barberá¹ *, Juan Ángel García-Pardo¹, Rafael Ramos-Garijo¹
Alberto Palomares², Vicente Julián¹, Miguel Rebollo¹, and Vicent Botti¹

¹ Information Systems and Computing Department,
Universidad Politécnica de Valencia, 46022 Valencia, Spain
{sheras, jgarciapardo, rramosgarijo}@dsic.upv.es

² TISSAT S.A., Parque Tecnológico,
Av. Leonardo Da Vinci, 5, 46980 Paterna - Valencia, Spain
{apalomares}@tissat.es

Abstract. In this paper, a new CBR system for Technology Management Centers is presented. The system helps the staff of the centers to solve customer problems by finding solutions successfully applied to similar problems experienced in the past. This improves the satisfaction of customers and ensures a good reputation for the company who manages the center and thus, it may increase its profits. The CBR system is portable, flexible and multi-domain. It is implemented as a module of a help-desk application to make the CBR system as independent as possible of any change in the help-desk. Each phase of the reasoning cycle is implemented as a series of configurable plugins, making the CBR module easy to update and maintain. This system has been introduced and tested in a real Technology Management center ran by the Spanish company TISSAT S.A.

1 Introduction

Technology Management Centers (TMCs) are control centers in charge of managing all processes implicated in the provision of technological and customer support services in private companies and public administration organisms. Usually, the company managing the TMC also has a call center, where a group of operators attend to requests of customers with the help of a help-desk software. The call center is also an effective way to communicate government organisms and citizens. Therefore, the operators of the call center must deal with queries coming from very diverse domains.

Nowadays, differentiating a company from its competitors in the market just by its products, prices and quality is becoming very difficult. Thus, companies try to take advantage by a high-quality customer support. A big amount of commercial activity is performed via phone, being necessary to avoid situations as

* Financial support from Spanish government under grant PROFIT FIT-340001-2004-11 is gratefully acknowledged.

busy lines, to ask the customer to repeat the query several times or to give incoherent answers. Moreover, a good customer support depends on the experience and skills of the company operators. There is an obvious need for saving their experience and for giving a suitable answer to each query as quick as possible.

From the 90s, Case-Based Reasoning (CBR) systems have been used to cope with this need in help-desks applied to call centers [1][2][3][4][5][6][7]. More recently, the internal CAD/CAM help-desk system *Homer* [8][9] has been developed in the course of the *INRECA-II* project [10]. There are also many companies that sell software tools for applying CBR to help-desks (e.g. *eGain* [11], *Kaidara* [12] and *Empolis* [13]). Therefore, most of the systems reported have either been specifically adapted to cover the needs of a private company by using some CBR tool [14][15][16] or developed for research purposes.

We were asked to implement a CBR system with specific features and observing some constraints. On one hand, the CBR system had to be flexible and modular, in order to be easily integrated in an existing help-desk application as an intelligent module for advising solutions to customer requests. On the other hand, we were not allowed to use any CBR tool that is only available for research purposes or copyrighted by any vendor. Therefore, we considered to implement a new CBR system able to fulfil these requirements.

The rest of the paper is structured as follows. In section 2 we briefly introduce the environment where our CBR system has been introduced. In section 3 we explain the CBR module proposed. In section 4 we show the results of the tests performed over the system. Finally, we summarise the conclusions of this paper.

2 I2TM - Intelligent and Integrated Ticketing Manager

The Spanish company TISSAT S.A. [17] runs a Technology Management Center (TMC) that offers customer support, communication and Internet services for public administration organisms and private companies. TISSAT works either with problems related to computer errors or with other domains, such as the international emergency phone 112 of Valencia (Spain), which covers the emergencies of over four and a half million of citizens.

TISSAT attends to customer requests via a call center. This call center can receive queries via phone, e-mail, Internet or fax. There is a maximum time to provide a correct solution for each query. This time is agreed between TISSAT and its customers in the Service Level Agreements (SLA's). When the maximum time to solve a problem is exceeded, the company is economically penalized.

In order to efficiently manage its call center, TISSAT has developed a help-desk toolkit called I2TM (**I**ntelligent and **I**ntegrated **T**icketing **M**anagement). I2TM manages customer requests, integrates the available channels to make a request and manages the inventory. The system also helps operators to solve new problems by searching for solutions successfully applied to similar problems in the past. This will ease their work and thus, they will be able to provide quicker and more accurate answers to customer problems. In order to cope with this functionality, we have developed a tool called CBR-TM (**C**ase-**B**ased **R**easoning

for Ticketing Management). This tool works as a separate module of the I2TM system, which allows to make changes in the I2TM implementation without affecting the CBR-TM module and vice versa. Figure 1 shows the overview of the entire system. I2TM and CBR-TM communicates and synchronises their data via webservice calls. The CBR-TM module will be explained in detail in section 3.

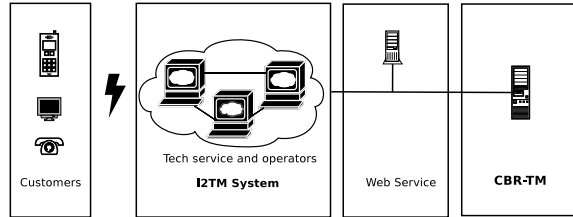


Fig. 1. System architecture

3 CBR-TM - CBR for Ticketing Management

Before the implementation of the CBR-TM module and the new I2TM system itself, some weaknesses to improve in the call center operation were identified. On one hand, it was necessary to save the knowledge and experience of the operators in an appropriate format (previously it was simply written in hand-written notes or in reference manuals that were usually out of date). This would avoid losing valuable information whenever the operators leave the company and it may also be used to train new operators. Moreover, the information about problems that had been already solved by other operator was not available on-line and the operators lost time solving them again. On the other hand, the information to manage comes from a wide range of domains and data types.

In order to facilitate the update of the CBR-TM module, each phase of the reasoning cycle [18][19] (*Retrieve, Reuse, Revise* and *Retain* [20]) is implemented as a *plugin algorithm*. Thus, CBR-TM is a flexible system and any change in the algorithms that implement the phases, or even the introduction of new algorithms, does not affect the entire CBR-TM system. The specific algorithm that has to be used in each phase is specified in a *XML configuration file*. The following sections describe with more detail the reasoning phases of the CBR-TM module.

3.1 Data acquisition

An important task in this project has been to obtain a test database to validate our CBR system during its development. In order to extract this information, we analysed the old call center database. The registers of the database (*tickets*)

contain information about previously solved problems. Therefore, a *ticket* in our system is a new case to solve. The data structure in CBR-TM and the relations with the structure of the new databases of I2TM is shown in Figure 2.

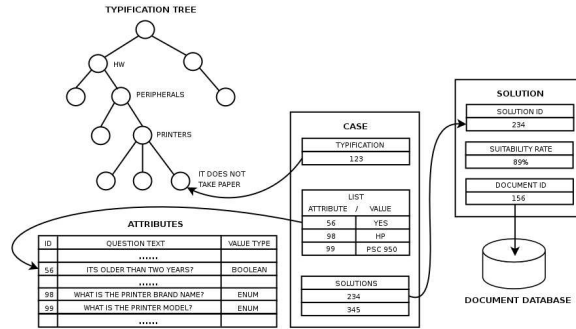


Fig. 2. Overview of the data structure in I2TM and CBR-TM

TISSAT maintains a non-disjoint tree (*Typification Tree*) that contains the taxonomy of the problem types (categories) in a hierarchical order (from less to more specific categories). These categories are set by TISSAT depending on the application domain of each project managed by the company. The first level nodes of the tree represent projects and the nodes below them are the categories of those projects. The CBR-TM module is able to reread the tree whenever a new project is added or any category is modified. In this sense, CBR-TM is a multi-domain system able to work with different types of problems. TISSAT also maintains a database of answers to questions that the operators ask to the customer when a query is made. These answers are saved as *attributes* in a database and they provide more specific information about the problem represented by the categories. In addition, TISSAT registers successfully applied solutions in a *document database*. In CBR-TM, a case is the prototyped representation of a set of tickets sharing the same categories and attributes. Each case has one or more associated solutions. One solution of the document database can also be associated with more than one case. CBR-TM stores the cases in a *case-base*.

3.2 Retrieve

The first step when CBR-TM is asked to solve a new ticket is to retrieve a set of cases from the case-base that are related to the same problem as the ticket. I2TM uses a webservice call named *GetSolutions* to start this process in the CBR-TM module. The call needs as parameters the values of the ticket attributes and its categorisation. The retrieval process comprises three steps: *Indexation*, *Mapping* and *Similarity* calculation. At the end of the retrieval phase, a list of cases sorted by similarity with the ticket is obtained. This phase is implemented through three different types of plugin algorithms: the *Indexer*, the

Mapper and the *Similarity* algorithms. The *Indexer* algorithm hierarchically organises the cases of the case-base in order to facilitate their retrieval. Currently, the operators perform the indexation by categorising manually the ticket. The *Mapper* algorithm explores the *Typification Tree* to retrieve the category nodes of the ticket and its predecessors (since upper categorisations represent more generic problems, but they are also related with the current problem and their solutions might also be suitable). Then, the algorithm searches in the case-base and retrieves all the cases with either the same categorisation as the ticket or a more generic one.

Once the set of similar cases has been selected, it is sorted by similarity with the ticket. The *Similarity* algorithm performs this arrangement. Here arises the problem of finding the similarity between cases that share some attributes and have different ones. Note that the cases associated with different categories of the *Typification Tree* can have different attributes. Moreover, there are many possible attribute types. The attributes can also have missing values, which makes more complicated the calculation of the similarity between cases. In order to test the CBR-TM module, we have adapted and implemented some similarity measures: two similarity measures based on the Euclidean distance (classic *Euclidean* and *NormalizedEuclidean*) and a similarity measure based on the ratio model proposed by *Tversky* [21]. In addition, we have implemented a set of distance metrics that allow us to work with different attribute types (numeric, nominal and enumerated). The *Similarity* algorithms use the distance metrics to compute local distances between the attributes of the cases, and the similarity measures to compute global distances between the cases (the similarity between the cases). Finally, the set of retrieved cases is sorted by means of a *k-nearest neighbour* algorithm.

3.3 Reuse

The reuse phase is implemented by means of the *SolutionSelection* plugin algorithm. At the end of the reuse phase, we obtain a sorted list of solutions to apply to the ticket. First, the *SolutionSelection* algorithm proposes the solutions of the most similar case to the ticket, sorted from higher to lower degree of suitability. Next, it proposes the solutions of the second most similar case, and so on. Note that the solutions themselves are not adapted, but proposed directly in a specific order to use them to solve the current ticket. When this process is finished, CBR-TM answers the *GetSolutions* webservice call and returns it with the list of proposed solutions and their associated suitability for the ticket.

3.4 Revise

In the revision phase, the I2TM system uses the *CloseQuestion* webservice call to report to the CBR-TM module the customer degree of satisfaction with the proposed solution. The tickets that were not requested to CBR-TM, but solved directly by the operator, are also reported. This phase, implemented by means of the *Rewarder* plugin algorithm, helps CBR-TM to improve its performance.

When CBR-TM is reported a solved ticket, it performs the retrieval phase in order to discover whether this ticket has already a prototype case in the case-base. If such case exists and the solution applied to the reported ticket is already associated with this case, the degree of suitability of this solution is increased. Otherwise, the new solution is associated with the case. If there is not a similar enough case in the case-base, a new case with its solution is created. The similarity threshold has been found experimentally and it can be changed to any desired value.

Note that the retrieval phase would be avoided here if we were able to know which case was used to propose its solution to solve the ticket. Moreover, this solution could be penalized if it does not fit the ticket. However, we consider that in the current implementation of our system this is not appropriate. On one hand, the CBR-TM module may be reported a ticket that was not requested previously to the module. In this situation we have to perform the retrieval phase in order to check if there is a similar case in the case-base or, otherwise, to create a new one. On the other hand, it is possible that CBR-TM had proposed an invalid solution but it had not made any mistake, since this is not a completely automated system and, for instance, the operators can fail in their categorisations. Moreover, do not use a proposed solution does not necessary mean that this solution is erroneous, but the operator may have chosen other solution for any reason.

3.5 Retain

As it is explained above, each time that a ticket is solved, the I2TM system reports back to the CBR-TM module. The retention phase is also done by means of the *Rewarder* algorithm, which checks if it is necessary to create a new prototype case for the ticket. Therefore, the retention phase can be viewed as a consequence of the revision phase. If the ticket that has been reported to CBR-TM is not similar enough to any case of the case-base (it exceeds the similarity threshold that has been specified), a new case will be added to the case-base.

4 Evaluation

Using the *Ticket Database*, we have run several tests to validate the CBR-TM module. The tests have been performed using a cross-partition technique, separating the ticket database into two databases for training (loading the case-base) and testing the system. We wanted to check on the computer error domain the behaviour of the similarity measures implemented. Therefore, the tests have been repeated setting the system to work with a different similarity measure each time.

First of all, we have checked the system performance. This performance may be influenced by the size of the database or by the number of customers performing simultaneous requests. Figure 3a shows that as the number of tickets in the database used to create the case-base of CBR-TM increases the mean error in the answers to the requests decreases. Note that, as we are performing a supervised

learning, it is considered an error when CBR-TM does not propose the same solution as the one we have recorded in the *Ticket Database* for the ticket that has been requested. It demonstrates that, the more problems CBR-TM solves, the more it increases its knowledge to solve new ones.

Figure 3b shows the response time of the CBR-TM module when the number of customers performing simultaneous requests increases. Although in this test it is considered that the customers are making the requests almost at the same time, CBR-TM is able to answer all of them quickly. With regard to the behaviour of the different similarity measures, we can appreciate that their performance in this domain is almost the same.

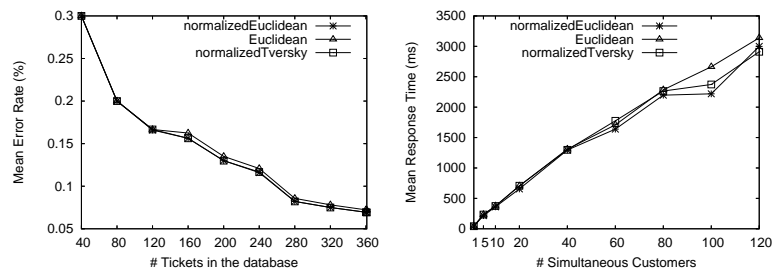


Fig. 3. a: Influence of the database size on the CBR-TM system performance; b: Influence of the number of simultaneous customers on the CBR-TM system performance

5 Conclusions

We have developed a CBR system for ticketing management called CBR-TM, which acts as an intelligent module for the I2TM help-desk application in the Spanish company TISSAT S.A. The CBR-TM module searches for solutions successfully applied in the past and thus, helps the operators to rapidly solve new problems. This saves time and prevents I2TM from losing the knowledge acquired when a problem is solved. The results of the CBR-TM evaluation show that the system has a good performance when it attends to the requests of simultaneous customers. As it is expected, the CBR-TM accuracy improves as the case-base increases and the system learns properly the new solutions created by the I2TM operators.

The system has been tested in a help-desk whose purpose is to solve computer errors, but TISSAT is planning to apply it to other domains. The system is recently implanted and an intensive research to improve the techniques applied will be done. One of the main objectives in a near future is to develop an automatic categorisation method, in order to prevent the CBR-TM module from human mistakes. Current research is done in studying automatic methods to set appropriate weights to the attributes of the cases and improve the similarity calculation.

References

1. Acorn, T., Walden, S.: SMART: Support Management Automated Reasoning Technology for Compaq Customer Service. In Scott, A., Klahr, P., eds.: Proceedings of the 2 International Conference on Intelligent Tutoring Systems, ITS-92 Berlin. Volume 4., AAAI Press (1992) 3–18
2. Simoudis, E.: Using Case-Based Retrieval for Customer Technical Support. *IEEE Intelligent Systems* **7**(5) (1992) 7,10–12
3. Kriegsman, M., Barletta, R.: Building a Case-Based Help Desk Application. *IEEE Expert: Intelligent Systems and Their Applications* **8**(6) (1993) 18–26
4. Shimazu, H., Shibata, A., Nihei, K.: Case-Based Retrieval Interface Adapted to Customer-Initiated Dialogues in Help Desk Operations. In Mylopoulos, J., Reiter, R., eds.: Proceedings of the 12th National Conference on Artificial Intelligence. Volume 1., Seattle, USA, AAAI Press (1994) 513–518
5. Raman, R., Chang, K.H., Carlisle, W.H., Cross, J.H.: A self-improving helpdesk service system using case-based reasoning techniques. *Computers in Industry* **2**(30) (1996) 113–125
6. Kang, B.H., Yoshida, K., Motoda, H., Compton, P.: Help Desk System with Intelligent Interface. *Applied Artificial Intelligence* **11**(7-8) (1997) 611–631
7. Roth-Berghofer, T., Iglezakis, I.: Developing an Integrated Multilevel Help-Desk Support System. In: Proceedings of the 8th German Workshop on Case-Based Reasoning. (2000) 145–155
8. Goker, M., Roth-Berghofer, T.: The development and utilization of the case-based help-desk support system HOMER. *Engineering Applications of Artificial Intelligence* **12**(6) (1999) 665–680
9. Roth-Berghofer, T.R.: Learning from HOMER, a case-based help-desk support system. In Melnik, G., Holz, H., eds.: *Advances in Learning Software Organizations*, Springer-Verlag (2004) 88–97
10. Bergmann, R., Althoff, K.D., Breen, S., Göker, M., Manago, M., Traphöner, R., Wess, S.: *Developing Industrial Case-Based Reasoning Applications . The INRECA Methodology*. 2nd edn. Volume 1612 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag (2003)
11. eGain: www.egain.com (2006)
12. Kaidara Software Corporation: <http://www.kaidara.com/> (2006)
13. Empolis Knowledge Management GmbH - Arvato AG: <http://www.empolis.com/> (2006)
14. Althoff, K.D., Auriol, E., Barletta, R., Manago, M.: *A Review of Industrial Case-Based Reasoning Tools*. AI Perspectives Report. Goodall, A., Oxford (1995)
15. Watson, I.: *Applying Case-Based Reasoning. Techniques for Enterprise Systems*. Morgan Kaufmann Publishers, Inc., California (1997)
16. empolis: empolis Orenge Technology Whitepaper. Technical report, empolis GmbH (2002)
17. Tissat S.A: www.tissat.es (2006)
18. Giraud-Carrier, C., Martinez, T.R.: An integrated framework for learning and reasoning. *Journal of Artificial Intelligence Research* **3** (1995) 147–185
19. Corchado, J.M., Borrajo, M.L., Pellicer, M.A., Yanez, J.C.: Neuro-symbolic system for Business Internal Control. *Advances in Data Mining, LNIA Springer-Verlag* **3275** (2004) 1–10
20. Aamodt, A., Plaza, E.: Case-based reasoning: foundational issues, methodological variations and system approaches. *AI Communications* **7**, no. 1 (1994) 39–59
21. Tversky, A.: Features of similarity. *Psychological Review* **84**, no.4 (1997) 327–352