
A survey of distributed and data intensive CBR systems.

Aitor Mata

Departamento Informática y Automática
Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
aitor@usal.es

Abstract Case-Based Reasoning is a methodology that uses information that has been considered as valid in previous situations to solve new problems. That use of the information allows CBR systems to be applied to different fields where the re-utilization of past good solutions is a key factor. In this paper some of the most modern applications of the CBR methodology are revised in order to obtain a global vision of the techniques used to develop functional systems. In order to analyze the systems, the four main phases of the CBR cycle are considered as the key elements to organize an application based on CBR.

Keywords: Case-Based Reasoning, recover, reuse, revise, retain.

1 Introduction

Case-Based reasoning (CBR) is one of the paradigms developed by artificial intelligence in order to build intelligent knowledge-based systems. CBR has received a great deal of attention in recent years and has been successfully used in a great variety of areas.

CBR systems are normally structured following the classic CBR cycle. That structure is composed of four main phases: *retrieval*, *reuse*, *revision* and *retention*. Those four phases are a way of organize the process used to solve problems by human beings. This paper follows the CBR cycle structure with the aim of explain the modern approaches to the methodology used to face different problems, and concentrates in problems with big amount of data or distributed applications.

2 Case Based Reasoning Systems

CBR system requires efficient techniques for several important subtasks, such as organizing and maintaining the case base, retrieving cases (which are maximally similar to the problem) from the case base, and adapting stored cases to the problem at hand. The basic inference mechanism underlying CBR, where the concept of similarity plays a major role, is built upon the principle of instance based learning and nearest neighbor classification.

2.1 Case definition and case base creation

The first steps in the design of a CBR application must consist in a transformation of the information available into a structure, into *cases*. In textual case bases, it is sometimes necessary to extract knowledge from the data before creating the case base [1]. Once the knowledge is obtained, it can be structured into the case base. Every new element is part of one or more of the *pieces of knowledge* previously identified, and then, the case is formed by the separated pieces that has inside it.

In medical applications, the case must include values referred to the patient, but also associated with the clinical evolution of the patient [2]. It is also interesting to include a *reputation* value, which is increased every time a case is recovered from the case base and used, every time the expert considers that the case is useful.

When the information to be transformed into cases contains a great amount of words, it is necessary to parse the original data [3] in order to obtain the list of terms used to create the cases. In some occasions, the information can be considered as hard to model, but after an analysis, it can be transformed into numerical variables [4] with what is quite easier to generate cases.

2.2 Recovering the data from the case base

When a new problem appears, a selection of cases are recovered from the case based and will be used to solve that new problem. If the amount of variables is quite big, it is necessary to select which ones will be used to select the similar cases from the case base [2]. A two steps procedure occurs so first the interesting variables must be chosen, and then, the search in the case base of the most similar cases according to those variables. If different features are considered when defining the case base, they must all be considered when obtaining similar cases from the case base. In this kind of situations different metrics can be done to calculate the similarity of the different features [4], and then create a combined similarity metric that integrates all the metrics used.

In some circumstances, a previous search of context is done [5], to obtain a variety of cases that are used to perform a second and more specific search.

If the problem introduced in the system implies considering different scenarios, multiple retrievals can be done [6]. In this kind of situations the original problem introduced in the system defines the start point of the search and from that point and looking for in different directions, different sets of cases are recovered from the case base, in order to generate a complete perspective of the problem.

2.3 Adaptation of the retrieved cases

Transforming the recovered cases from the case base is the solution generator for the CBR systems. From the collection of cases retrieved from the cases base, a new solution must be generated in order to solve the proposed problem. Numeric situations, like those used in microarray problems, can be reused thru neural networks like Growing Cells Structures [7], where the aim is to cluster the retrieved information. Another way to use neural networks to adapt the retrieved information is to change the weight of the connection between the neurons depending on the retrieved cases [8]. Changing the weights allows the system to adapt the solution to the problem, as the retrieved cases will depend directly on the proposed problem.

If the problem to be solved may belong to more than one field of knowledge, and there may be more than one case base, a good solution can be to adapt the retrieved cases, from the different case bases, according to the characteristics of the problem [9]. In this case, neural networks were used to recover the data from the different case bases, and machine learning algorithms combined the retrieved cases in order to adapt those cases to the proposed problem.

When using genetic algorithms, the reuse may help to reduce convergence time if considering previously working solutions [10]. This approach may be applied to different fields where evolutionary algorithms are useful but slow.

2.4 Review of the proposed solution

When a solution is generated by a CBR system, it is necessary to validate the correction of that solution. This may be the most complicated phase of the cycle. In crucial fields, such as medical applications, it is normal to trust an expert in order to finally accept a solution [11]. Then, after being accepted by the corresponding expert, next time it will be considered as a better solution, being chosen from the case base with a higher probability.

Changing the values proposed by the system to others *similar* but not equal is a technique also used to revise the correction of a solution [12]. If the solution generated by the similar values is not better than the proposed one, then the chosen

one is a good solution for the problem. Genetic algorithms are also used to revise the correction of the solutions [13]. After running those algorithms, the solutions can be accepted, and added to the case base.

2.5 Storage of the solution and maintenance of the case base

In most cases there a big amount of information is stored in the case base and it is not necessary to store every valid case, thus the information could be too redundant. In those situations a conditional retention is performed [14], keeping the new solution only if it is different enough to the closest existing case.

Even when the proposed solution is considered as an eventually good solution to be stored in the case base, the growth of the case base can be counterproductive. In some case, where the amount of stored information is huge and when there must be an economy of resources in order to manage a reasonable case base, case base editing is necessary [15].

When the case base grows to thousands of elements, it may be difficult to maintain it. Then dividing the case base in different parts with certain inner similarity [16] can help to structure the store information and also to make future retrievals. Another strategy used to control the growth of the case base is to group cases into prototypes that include the common characteristics of a series of cases with no plenty of variability.

Table 1. Resume of strategies used in the four main phases of the CBR cycle.

Phase	Strategies
<i>Case Definition</i>	<ul style="list-style-type: none"> ○ Numerical list of parameters ○ List of abstract concepts extracted from textual information
<i>Retrieval</i>	<ul style="list-style-type: none"> ○ Metrics ○ Two steps search
<i>Reuse</i>	<ul style="list-style-type: none"> ○ Collaboration between algorithms ○ Transforming neural networks
<i>Revision</i>	<ul style="list-style-type: none"> ○ Experts ○ Alternative structures
<i>Retention</i>	<ul style="list-style-type: none"> ○ Eliminate redundant data ○ Division of the case base

3 Resume of solutions

After analyzing the main phases of the CBR cycle, it is necessary to clearly resume the various techniques used to improve the capabilities of each one, de-

pending, in most of the cases, on the problem to be solved or on the type of data managed. Retrieval and reuse are the two phases where more improving efforts have been expended. Those are the phases where the quality of the information is checked. *Table 1* shows a resume of the solutions used in every phase.

References

1. Mustafaraj, E. (2007) Knowledge Extraction and Summarization for Textual Case-Based Reasoning, *Philipps-Universität Marburg*.
2. Montani, S. (2007) Case-based Reasoning for managing non-compliance with clinical guidelines. *International Conference on Case-Based Reasoning, ICCBR 2007, Proceedings*.
3. Patterson, D., Rooney, N., Dobrynin, V. and Galushka, M. (2005) Sophia: A novel approach for Textual Case-based Reasoning, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*.
4. Ros, R., Veloso, M., de Mantaras, R.L., Sierra, C., et al. (2006) Retrieving and Reusing Game Plays for Robot Soccer, *Advances in Case-Based Reasoning*, 4106, 2006.
5. Spasic, I., Ananiadou, S. and Tsujii, J. (2005) MaSTerClass: a case-based reasoning system for the classification of biomedical terms, *Bioinformatics*, 21 (11), 2748-2758.
6. Aha, D.W., Molineaux, M. and Ponsen, M. (2005) Learning to Win: Case-Based Plan Selection in a Real-Time Strategy Game. In, *Case-Based Reasoning Research and Development*. 5-20.
7. Diaz, F., Fdez-Riverola, F. and Corchado, J.M. (2006) Gene-CBR: A case-based reasoning tool for cancer diagnosis using microarray data sets, *Computational Intelligence*, 22 (3/4), 254-268.
8. Zhang, F., Ha, M.H., Wang, X.Z. and Li, X.H. (2004) Case adaptation using estimators of neural network, *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, 2004, 4.
9. Policastro, C.A., Carvalho, A.C. and Delbem, A.C.B. (2006) Automatic knowledge learning and case adaptation with a hybrid committee approach, *Journal of Applied Logic*, 4 (1), 26-38.
10. Pérez, E.I., Coello, C.A.C. and Aguirre, A.H. (2005) Extraction and reuse of design patterns from genetic algorithms using case-based reasoning, *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 9 (1), 44-53.
11. Chang, C.L. (2005) Using case-based reasoning to diagnostic screening of children with developmental delay, *Expert Systems With Applications*, 28 (2), 237-247.
12. Li, H., Hu, D., Hao, T., Wenyin, L., et al. (2007) Adaptation Rule Learning for Case-Based Reasoning, *Semantics, Knowledge and Grid, Third International Conference on*, 44-49.
13. Pavón, R., Díaz, F., Laza, R. and Luzón, V. (2008) Automatic parameter tuning with a Bayesian case-based reasoning system. A case of study, *Expert Systems With Applications*.
14. Sharma, M., Holmes, M., Santamaria, J., Irani, A., et al. (2007) Transfer learning in real-time strategy games using hybrid cbr/rl, *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*.
15. Delany, S.J. (2006) Using Case-Based Reasoning for Spam Filtering, *Dublin Institute of Technology*.
16. Li, J.Y., Ni, Z.W., Liu, X. and Liu, H.T. (2006) Case-Base Maintenance Based on Multi-Layer Alternative-Covering Algorithm, *Machine Learning and Cybernetics, 2006 International Conference on*, 2035-2039.