

Case-Based Reasoning – A Short Introduction

Christiane Gresse von Wangenheim

1. Introduction

Case-Based Reasoning (CBR) [BBL⁺98,Wes95,AP94,Kol93] is an approach to develop knowledge-based systems that are able to retrieve and reuse solutions that have worked for similar situations in the past.

CBR is a problem solving paradigm that in many respects fundamentally differs from other Artificial Intelligence approaches [AP94]. Instead of relying solely on general knowledge of a problem domain or making associations along generalized relationships between problem descriptors and conclusions, CBR is able to utilize the specific knowledge of previously experienced, concrete problem solutions. A new problem is solved by finding a similar past case and reusing it in the new problem situation (see Figure 1). A second important difference is that CBR is an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems.

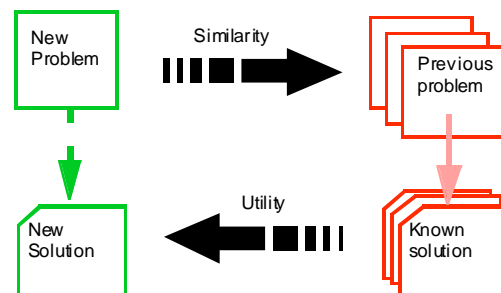


Figure 1. Basic idea of CBR approach

CBR is based on a model of human cognition dealing with knowledge in form of concrete experienced examples. It arises out of research in the area of Cognitive Science basically in the work of Schank and Abelson on dynamic memory and the central role that a reminding of earlier situations (episodes or cases) and situation pattern (in form of scripts describing information about stereotypical events and Memory Organization Packages (MOPs), expressing the situation patterns) have in problem solving and learning.

Also research in the area of analogical reasoning and from theories of concept formation, problem solving and experiential learning within philosophy and psychology. As Wittgenstein observes, concepts

are polymorphic and their instances can often be categorized in a variety of ways, so that it is impossible to come up with one useful classification. A solution therefore is the representation of a context extensionally through its set of instances (or cases).

A case typically represents the description of a (problem) situation together with the experiences gathered during the solution of the (problem) situation. It may also contain other items such as the effects of the solution applied or a justification for the solution and explanation or be enriched by an administrative part (including e.g., a case number). A case can be defined as a contextualized piece of knowledge which records an episode where a problem or problem situation was totally or partially solved. Cases primarily contain concrete experiences experienced in a specific situation. However, they can also be lifted to abstract cases, subsuming the experience described in a set of concrete cases.

In order to be available to be reused, cases are organized and stored in a case base. In addition to the case base, a CBR system may include some general knowledge in the form of models or rules or constraints. The case base and the general knowledge constitute a partial domain model of the domain of application.

2. The CBR Cycle

A widely accepted model of the CBR process is the CBR cycle proposed by Aamondt and Plaza [AP94] which comprises four principle tasks (see Figure 2): retrieve, reuse, revise, and retain.

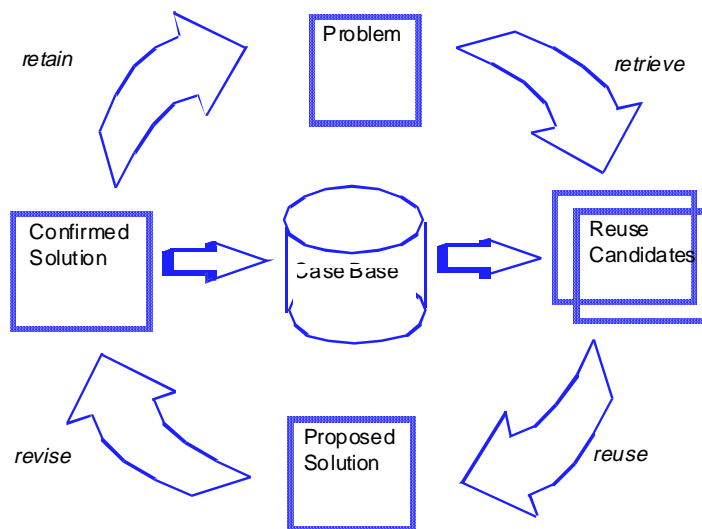


Figure 2. CBR-cycle (4re)

In the center of the cycle is set of cases reporting previous experiences stored in the case base and, possibly, additional general knowledge of the domain. General domain knowledge is applied during different steps of the CBR process and provides, for example, control knowledge for the consistency of the cases, the search of similar cases or the adaptation of them regarding the new problem.

During the retrieval the most similar case or a set of cases in the case base is determined, based on the new problem description. During reuse the information and knowledge in the retrieved case(s) is used to solve the new problem. During revision the applicability of the proposed solution is evaluated in the real-

world. If necessary, the proposed case is adapted in some way in order to completely fulfill the needs of the present situation. If the case solution generated during the revise phase should be kept for future problem solving, the case base is updated with a new learned case in the retain phase.

A fundamental idea is that those four tasks are used in a continuous reasoning cycle. The CBR cycle is further decomposed in a hierarchy of CBR tasks, denoted as task-method decomposition [AP94] (see Figure 3).

While this is a very general model, a number of variations exist [Wat97,Alt96,AP94,Kol93]. Each of the subtasks may be implemented through a variety of techniques, as well as, the representation of cases highly varies and depends on the specific application domain of the CBR system.

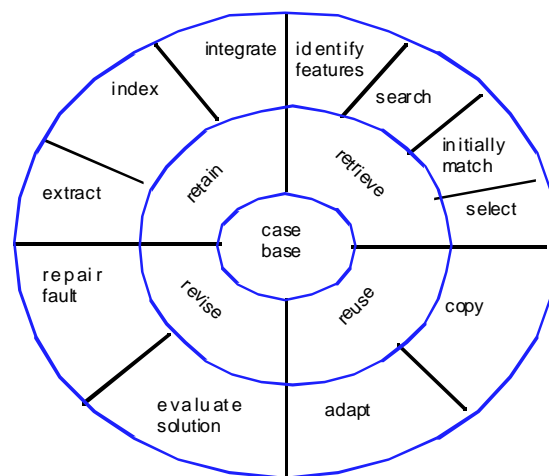


Figure 3. Task decomposition model

Case Base and Case Representation

The case base is a collection of cases, representing specific problem solving experiences tied to a context in which the experience can be used. A case records knowledge on an operational level. Although cases can vary greatly in form and size depending on the application domain, the content of a case typically includes a lesson it teaches and the context in which the lesson can be used [Kol93]. The context description indicates under which circumstances it is appropriate to retrieve the case. Basically, a case contains [Kol93]:

- problem/situation description: describing the state of the world when the case occurred and, if appropriate, what problem was needed to be solved at that time.
- solution: describing the solution to the problem specified in the problem description or the reaction to the given situation.
- outcome: the resulting state of the world when the solution was applied.

However, the concepts of problem and solution have no general definitions and vary from application to application. The problems of interest can range from very specific tasks, such as classification, to quite general situations as in decision support. For the former, the notion of a solution is well defined while this is not the case for the latter. In classification tasks, the solution is the class an object belongs to, for a diagnostic problem the solution is a diagnosis. In case of decision support tasks a solution may also be an

action, plan or just a useful piece of information for the user.

In a CBR-system, cases can be represented in several ways. The most simple form is the description of

the application focused by this research work, we do not discuss these techniques here in detail.

The concept of similarity is the key notion to realize inexact matching. Various approaches exist for the determination of the similarity. One, commonly used, is through similarity measures. A similarity measure assigns a numerical value to the case, expressing its degree of similarity with the given query, inducing a partial ordering on the set of problem descriptions and, consequently, also on the case base. These measures are often based on a geometrical interpretation, where the cases are considered in a n-dimensional space. Each dimension corresponds to one attribute of the problem description of the cases. The similarity between two problem descriptions, which are represented as two points in an n-dimensional space is reduced to the geometric distance. Depending on the specific application domain and the knowledge representation, a number of standard measures exist which can be used when implementing a CBR system [Alt96,Wes95].

Let CBP denote a set of input descriptions P for which a solution S exists such that (P,S) is in the case base CB. A **similarity measure** is a mapping [Ric99]:

$$\text{sim}: P \times \text{CBP} \rightarrow [0,1] \in \mathbb{R}$$

In order to reduce arbitrariness some assumptions are common (but not necessarily generally assumed):

- (i) $0 \leq \text{sim}(x,y) \leq 1$ (normalization)
- (ii) $\text{sim}(x,x) = 1$ (each object is itself the nearest neighbor)
- (iii) $\text{sim}(x,y) = \text{sim}(y,x)$ (symmetric property)
- (iv) $d(x,z) \leq d(x,y) + d(y,z)$ (triangle inequality)

For an attribute-value representation, a simple similarity measure is the generalized hamming measure that combines the importance of each attribute of the problem description with its local similarity value and sums the values to create a global similarity value for each case:

$$\text{sim}(P_1,P_2) = \sum w_j * \text{sim}_j(P_1,P_2)$$

sim_j : local similarity for attribute j;

w_j : relevance (weight) of attribute j for the problem description

If problems are coded as n-dimensional real vectors, classical mathematical metrics such as the Euclidean or Manhattan distance are often used. Another well known approach is based on a set theoretical model of Tversky, the contrast model. Here, the similarity of two objects or events is expressed through a linear contrast of weighted differences, which exist between their common and varying attributes. In this model, similarity is a function which increases in dependence on the number of common attributes and decreases in dependence on the number of different attributes. A refined model based on the contrast model has been applied in PATDEX [Wes95].

A similarity measure is a container which can store more or less sophisticated knowledge about a problem class. Local similarity deals with the similarity of values of a single attribute of the problem description. The local measures should represent domain knowledge. The global similarity measure are intended to express the usefulness of a case and depend on the pragmatics and are ultimately determined by the specific application. Global measures can be derived from local measures in various ways. The relative importance of attributes can be reflected by weights but also the relative position in a hierarchy as well as general background knowledge. Importance of an attribute to match depends on its overall impact with respect to achieving the reasoners current goal and its specific impact in individual cases.

Based on the similarity value determined for the cases, a best match is chosen. The best matching case(s) are usually determined by evaluating the degree of initial match more closely. The selection process typically generates consequences and expectations to justify non-identical attributes. This may

be done based on general domain knowledge or by asking the user for confirmation and additional information.

Reuse the Case(s) to Attempt to Solve the Problem

Once a matching case is retrieved, a CBR system attempts to reuse the solution suggested by the retrieved case in order to derive the solution for the new (problem) situation by adapting the retrieved case if necessary to completely fulfil the requirements of the present situation.

The simplest way to use a retrieved case is simply to copy the unchanged solution of that case as the solution to the actual problem. However, when having retrieved cases similar to a given query, these cases will hardly ever be directly applicable due to differences between the query and the problem description of the case. Hence, the solution that had been applied to the cases have to be adjusted accordingly through an adaptation process that takes into account those differences. The adaptation process includes the identification of differences between the retrieved case and the current situation and the transformation of the old solution into a solution for the new situation. Adaptation in simple situations can be achieved by, e.g., parameter adjustment according to rules and formulae or configuration methods according to rules and constraints. This requires additional domain knowledge which can be represented in form of rules. The extent of the adaptation performed by the CBR-system depends on the application task. Concerning knowledge management, where the objective is rather to provide intelligent assistance than generate new solutions, the reuse-phase consists mainly on the application of the suggested reuse candidate without modifications automatically done by the CBR systems. Necessary adaptations are done manually through the user, as these also, in order to be done intelligently, require complex domain knowledge, which in many application domains is lacking or would be too costly to be captured. Therefore, adaptation techniques are not further described here.

Revise the Proposed Solution

As CBR, due to the inexact matching, only suggests solutions, there may be a need for a correctness proof or an external validation. The objective of the revise phase is the evaluation of the created solution. The review of the solution can be performed by an expert, through a simulation or through its application in a real environment. If the revision has been successful, it is retained in the case base. When a case solution generated in the reuse phase is not correct an opportunity for learning from failure arises. Then the case solution is repaired using domain knowledge or user input. Case repair involves detecting the errors of the current solution and retrieving or generating explanations for them. The failure explanations are used to modify the solution in such a way that failures do not occur.

Retain the New Experiences as a New Case in the Case Base

Retainment is the process of incorporating what is valuable to retain from the new problem solving episode into the existing case base in order to make the knowledge available for future reuse. This step completes the experience feedback loop that is a necessary prerequisite for enabling a system to learn from experiences.

The learning from success or failure of the proposed solution is triggered by the outcome of the evaluation and possible repair. It involves selecting which information from the case to retain, in which form to retain, how to index the case for later retrieval, and how to integrate the new case in the case base structure.

Learning takes place from the given feedback to the overall system in order to improve a certain performance using some experience or instructions. Learning can appear in various ways, for example through inductive inference. Induction derives or improves a general solution method based on presented examples of a problem together with their solutions. An example is the generation of decision trees from classified examples. Machine learning methods can be used in order to improve the knowledge container of a CBR system (adding, creating, deleting cases) of the similarity measure (adjusting weights) and of

the solution transformation (new adaptation rules), as well as, techniques from statistics and information theory.

References

- [Alt96] K.-D. Althoff. Evaluating Case-Based Reasoning Systems: The Inreca Case Study. Habilitation, University of Kaiserslautern, Germany, 1996.
- [AP94] A. Aamodt, E. Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. *AI Communications*, 17(1), 1994.
- [BBL⁺98] H. D. Burkhard et al. (eds.), *Case-Based Reasoning Technology from Foundations to Applications*, Springer Verlag, 1998.
- [Kol93] J. L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Francisco, California, 1993.
- [Ric99] M.M. Richter. Tutorial on Case-Based Reasoning. Department of Computer Science, University of Kaiserslautern, Germany, 1999.
- [Wat97] I. Watson. *Applying Case-Based Reasoning Techniques for Enterprise Systems*. Morgan Kaufmann Publisher, California, 1997.
- [Wes95] S. Wess. Fallbasiertes Problemlösen in wissensbasierten Systemen zur Entscheidungsunterstützung und Diagnostik. Ph.D. Thesis, University of Kaiserslautern, Germany, infix Verlag, 1995.