Also in Proceedings of *11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems,* eds A. P. del Pobil, J. Mira & M. Ali, Lecture Notes in Artificial Intelligence 1416, Vol. 2, pp517-523, Springer Verlag.

CBR: Strengths and Weaknesses

Pádraig Cunningham

Department of Computer Science
Trinity College Dublin
Ireland
Padraig.Cunningham@cs.tcd.ie

Abstract. There is considerable enthusiasm about Case-Based Reasoning as a means of developing knowledge-based systems. There are two broad reasons for this enthusiasm. First, it is evident that much of human expert competence is experience based and it makes sense to adopt a reuse-based methodology for developing knowledge based systems. The other reason is the expectation that using Case-Based Reasoning to develop knowledge based systems will involve less knowledge engineering than alternative 'first-principles' based approaches. In this paper I explore the veracity of this assertion and outline the types of situation in which it will be true. CBR is perceived to have this knowledge engineering advantage because it allows the development of knowledge based systems in weak theory domains. If CBR can work without formalising a domain theory then there is a question about the quality of solutions produced by case-based systems. This is the other issue discussed in this paper and situations where CBR will and will not produce good quality solutions are outlined.

1 Introduction to Case-based reasoning

The idea of Case-Based Reasoning is intuitively appealing because it is evident that much of human problem solving competence is experience based. People draw on past experience when solving problems and can readily solve problems that are similar to ones encountered in the past. More than that, if a problem can be solved by reusing a solution from a solved problem that is similar then the new problem may be processed without much in-depth analysis. This leads to the expectation that Case-Based Reasoning (CBR) can be based on shallow knowledge and that developing CBR systems can require less knowledge engineering (KE) that alternative techniques.

In recent years a standard strategy for CBR has evolved and this is most commonly expressed as the four R's first introduced by Althoff (1989) (see also Aamodt and Plaza, 1994). Once a case-base which capturies previous problem solving episodes has been established the components of CBR according to this structure are:-

Retrieve the most similar case or cases

Reuse the information and knowledge in that case to solve the problem

Revise the proposed solution

Retain the parts of this experience likely to be useful for future problem solving

In practice it is often difficult to distinguish between the **Reuse** and **Revise** stages and it might be best to think of this as a single **Adaptation** stage. Adaptation is somewhat controversial in CBR because, if CBR is to offer any knowledge engineering advantages these will be lost if substantial case adaptation is to be supported. So there are those that would argue that CBR is sensible when adaptation is minimal or non-existent (see Barletta, 1994 for instance). It is true that at the moment most successful installed CBR systems involve little adaptation.

The status of adaptation in CBR can best be explored in the context of the diagram shown in Figure 1. The motivation in CBR is to avoid the first principles reasoning that is required to find a solution (**SL**) to the problem specification (**SP**) in the target case. Instead a case with a similar specification (**SP`**) is retrieved from the case base and the solution to that case is adapted to fit the target problem. The expectation is that retrieval and adaptation will be simpler than the first principles reasoning. An alternative motivation may be that it is not possible to automate the first principles reasoning and instead the expectation is that the 'good quality' characteristics of **SP`** will survive the transfer to **SP** in the adaptation.

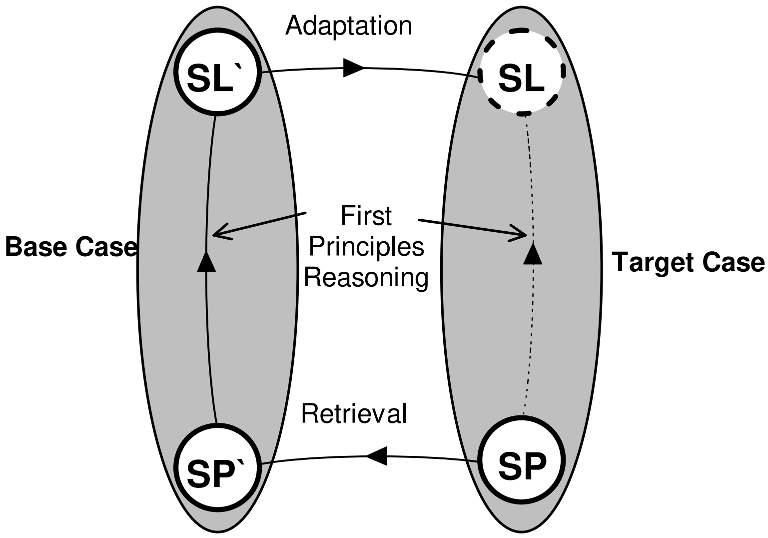

Fig. 1. The transformation and mappings involved in CBR: **SP** is a problem specification and **SL** is a solution.

This diagram also tells us something about the knowledge engineering requirements in CBR. CBR avoids having to encode the knowledge that would be required to automate the first principles reasoning. Instead the problem solving knowledge is encoded as follows:-

- In the case representation: the case representation must capture the predictive features of a problem.
- In the similarity measure: it must be possible to retrieve cases that are genuinely similar to the target problem.
- In the adaptation knowledge: the adaptation mechanism must be able to sensibly manipulate case solutions. (based on analysis presented in (Richter 1995)).

The last of the 4 Rs is **Retain** and it is receiving increased attention in recent years. The competence of a CBR system can increase over time as new cases are processed and added to the case base. This is a learning process because it results in an improvement in the competence of the system. As large scale CBR systems have been fielded in business and industry it has become evident that mechanisms for extending and updating the case-base are important. This is just part of the larger case-base maintenance issue. Case-base maintenance involves identifying existing cases that are out of date and identifying new cases that extend the competence of the system.

These issues of Maintenance and Adaptation are covered in the paper by Smyth (1998) and in the paper by Wilke and Bergmann (1998) in this volume. In the remainder of this paper I will cover the knowledge engineering aspects of CBR by addressing the twin questions; "Are knowledge based systems that are case-based comparatively easy to implement?” and “Can CBR produce top quality solutions?”. The final paper by Watson (1998) in this set of CBR papers presents some recent successful applications of CBR.

2 Case Representation & Retrieval

Before beginning our discussion of knowledge engineering and solution quality it is necessary to outline the standard approaches to representation and retrieval in CBR. In the Introduction it was established that cases have two components, the specification and the solution. The specification normally consists of a set of attributes and values. The attributes for a particular case should uniquely define that case and should be sufficient to predict the solution for that case. The case solution may simply be a single attribute representing a classification; for instance a fault category in diagnosis. Alternatively the solution may have complex structure as would be the case in a route planning system.

Retrieval is itself a huge area of research in CBR. However, commercial CBR tools normally offer the two alternatives of *k*-Nearest Neighbour (*k*-NN) and Decision Trees. *k*-Nearest Neighbour involves establishing a similarity metric by which the closeness of two cases can be measured. Then the target case is compared to all the cases in the case-base in turn and the *k* nearest are retrieved. This is a powerful technique if a good similarity metric can be established. It has the disadvantage that retrieval time increases directly with the size of the case-base.

The alternative is to use Decision Trees (*k*-D-Trees) where case retrieval is proportional to the depth of the D-Tree (i.e. *k*); the maximum depth is the number of attributes used in retrieval. Decision Trees can be built using induction if the case solution is an atomic category that can be used to drive the induction (Quinlan, 1986 & 1993). For CBR, *k*-D-Trees do have the disadvantage that they need to be rebuilt from scratch when new cases are added to the case-base.

There are CBR systems involving approaches to representation and retrieval that are radically different to this, however, the majority of CBR systems have specifications represented as feature vectors and use either *k*-NN or *k*-D Trees for retrieval.

3 Knowledge Engineering Requirements

In the introduction I have said that the enthusiasm for CBR is based on the following intuitive argument. Having access to similar problems in problem solving reduces the need for problem analysis - solution components from old problems can be reused and less in-depth analysis of the new problem is required. This suggests that developing CBR systems may require less knowledge engineering than, say, rule-based or model-based approaches.

This argument can be elaborated using the model shown in Figure 2. The key message in this diagram is that there are two stages in problem solving as formalised in knowledge based systems. The first stage involves an analysis of the problem that produces a representation of the problem that can be manipulated by the reasoning system - this representation is often a set of attribute values. The second stage involves developing the reasoning mechanism that manipulates the problem representation to produce a solution.

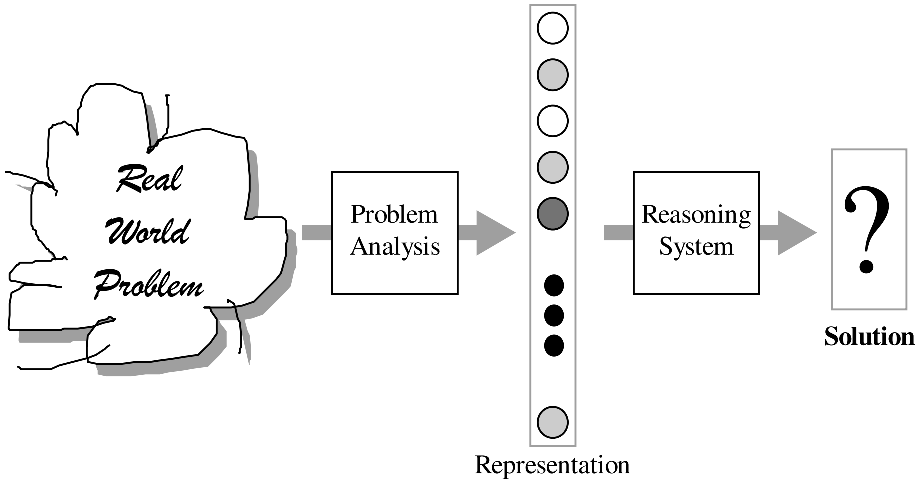

Fig. 2. A Model of knowledge based systems development.

For CBR this reasoning system is implemented as retrieval and adaptation components that are easier to develop than ‘from first principles’ reasoning mechanisms that require full knowledge engineering analysis of the problem domain.

If retrieval and adaptation mechanisms are easy to implement then CBR has clear KE advantages over ‘from first principles’ techniques. Many implemented CBR systems involve little or no adaptation and the reasoning mechanism is simply a retrieval system with solutions being used intact or with adaptation performed by the user. This advantage is less apparent if the problem analysis task that produces the problem representation dominates in the knowledge engineering effort. In the following sections we present two examples; one where this is not the case and one where it is.

3.1 BREATHALYSER

The BREATHALYSER system is an excellent example of the type of problem with which CBR can succeed much better than model based techniques. This system is a software system for predicting blood alcohol levels. BREATHALYSER requires the input of five features describing a subject and it produces an estimate of their blood alcohol level. It uses a case-base of approximately 100 cases gathered at pub closing time in Dublin. Some sample cases are shown in Figure 3.

N-1

Gender	Male
FrameSize	1
Amount	1
Meal	snack
Duration	60
BAC	0.2

N-3

Gender	Female
FrameSize	4
Amount	4
Meal	full
Duration	90
BAC	0.8

N-55

Gender	Male
FrameSize	1
Amount	3
Meal	snack
Duration	120
BAC	0.7

Fig. 3. Sample cases from the system.

For this case description to be complete there should also be an Elapsed-Time feature capturing the elapsed time since drinking ceased. In the circumstances where these cases were collected this feature is roughly constant at around one hour. So the system predicts blood alcohol levels where this holds. BREATHALYSER also supports some adaptation using rules that have been learned automatically from case base. When evaluated this system produced an accuracy of 90.5% using ‘leave-one-out’ testing (i.e. remove a case from the case-base and test to see what solution the remaining cases predict for that case.

BREATHALYSER is a good example of the system that was comparatively easy to develop in CBR. It would be difficult to develop using any strategy that would try to model interactions in the problem. CBR works well here because a workable set of predictive features is easy to identify so this knowledge engineering task that always has to be done even with CBR is easy in this situation.

3.2 The ISAC Experience

By contrast, the next system we describe proved difficult to develop even using CBR. ISAC is a CBR system for Air Traffic Control (Bonzano & Cunningham, 1996; Bonzano, Cunningham & Smyth, 1997). The objective for ISAC is to help the controller select a manoeuvre so as to resolve conflicts between aircraft. A conflict occurs when aircraft are on flight paths that cause them to pass too close. Developing a CBR system to address this problem has been difficult because the important features were difficult to determine and the relative importance of features were difficult to gauge. This problem has been confounded by the fact that the features are highly context sensitive; features that are very predictive in some conflicts are not relevant in others. In addition different controllers solve similar problems in different ways, solutions that are considered standard in one sector might be considered unusual in another sector.

There have been several attempts in the past to produce knowledge-based solutions to this problem. (e.g. (Shively & Schwamb, 1984)) These attempts have all tried to model the conflict resolution process and have met with mixed success. The CBR solution produced in ISAC has been demonstrated to work but the system took considerable time to develop. This is best explained with reference to the model of KBS development shown in Figure 2. In developing a KBS for this conflict resolution task; the problem analysis task that produces a problem representation represented a considerable part of the overall knowledge engineering effort. This contrasts with the situation in the BREATHALYSER example above where a case-representation was readily available.

While we would maintain that CBR provided a good solution to the conflict resolution problem, the advantages in development effort were not so great because the task of coming up with a good problem representation was considerable.

4 Solution Quality

The CBR methodology has a lot in common with that of example-based machine translation (EBMT). One of the claims made for EBMT is that it can produce very good quality translations because idiomatic structures get reused in EBMT and lend a ‘naturalness’ to the new translation. CBR can have the same type of advantage: high quality solution characteristics can be transferred to new situations through case adaptation in CBR. If these high quality solution characteristics are aspects of the problem that are difficult to formalise then this is an advantage that CBR has over, say, model based techniques.

An example of this is presented by Veloso in the current version of the Prodigy system (Veloso, 1994; Veloso 1997). Prodigy is a route planning system that is currently being used for planning routes for emergency vehicles in Pittsburg, USA. The surface characteristics of good solutions in this domain, such as distance and duration are easy to formalise. However there are other solution characteristics that are not so easy to formalise. For instance, it is important that the driver should be familiar with the route that is presented in a solution. It is probably impossible to formalise the driver’s level of comfort with a particular route. However, this aspect of solution quality is implicitly encoded in a good case and can transfer to new cases in adaptation.

In conclusion, where a strong domain theory is missing or difficult to formalise CBR can produce good quality solutions by adapting existing high quality solutions. If a domain theory is available then it will be possible to produce a good 'from first principles' reasoning system and this may produce better quality solutions in situations where solution quality gets 'damaged' in the adaptation process. Cunningham & Smyth (1997) have demonstrated this characteristic of CBR solutions in a case-based scheduling system.

5 Conclusions

Two important characteristics of CBR systems are the knowledge engineering requirements of system development and the quality of solutions produced. Knowledge engineering has two components, deciding on a problem representation and determining the causal interactions. CBR does not help with the first of these but can reduce or eliminate the second. If deciding on a problem representation does not dominate the knowledge engineering task then CBR can reduce KE requirements.

Case-based systems can produce good quality solutions in a weak theory domain where features affecting quality are difficult to formalise and solution quality can be carried implicitly in cases. However, if a good 'from first principles' system exists then it may beat the CBR system on quality in situations where solution quality gets lost by naïve adaptation methods.

References

Aamodt A., Plaza E., (1994) Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *Artificial Intelligence Communications*, Vol. 7, No. 1, Pp39-58

Althoff, K.D (1989) Knowledge acquisition in the domain of CNC machine centers; the MOLTKE approach. In John Boose, Brian Gaines, Jean-Gabriel Ganascia (eds.): *EKAW-89; Third European Workshop on Knowledge-Based Systems*, Paris, July 1989. pp 180-195.

Barletta, E.R., (1994) A Hybrid Indexing And Retrieval Strategy For Advisory CBRSystems Built With ReMind, 1994, *Proceedings of the European Workshop onCase-Based Reasoning.*

Bonzano A., & Cuningham P., (1996) ISAC: A CBR System for Decision support in Air Traffic Control *in Proceedings of EWCBR '96, Advances in Case-Based Reasoning*, Ian Smith & Boi Faltings eds. Springer Verlag Lecture Notes in AI, pp44-57.

Bonzano, A., Cunningham, P., & Smyth, B., (1997). Using introspective learning to improve retrieval in CBR: A case study in air traffic control, *in Proceedings of International Conference on Case-Based Reasoning,* Leake, D. & Plaza, E., (eds) Springer Verlag, pp413-424.

Cunningham, P., Smyth, B., (1997) Case-Based Reasoning in Scheduling: Reusing Solution Components, to appear in *The International Journal of Production Research*.

Quinlan J.R., (1986) Induction of Decision Trees, *Machine Learning*, 1, 81-106.

Quinlan J.R., (1993) *C4.5 Programs for machine learning*, Morgan Kaufmann Publishers

Richter, M. M., (1995). The knowledge contained in similarity measures*, Invited talk at ICCBR'95*.
http://wwwagr.informatik.uni-kl.de/~lsa/CBR/Richtericcbr95remarks.html.

Shively C., Schwamb K.B., (1984) AIRPAC: Advisor for the Intelligent Resolution of

Predicted Aircraft Conflicts, Mitre Corporation, MTR-84W164

Smyth, B., (1998) Case-Base Maintenance, *Proceedings of IEA-98-AIE, Lecture Notes in Computer Science,* Springer-Verlag, Berlin Heidelberg New York.

Veloso, M., (1997) Merge strategies for Multiple Case Plan Replay*, in Proceedings of International Conference on Case-Based Reasoning,* Leake, D. & Plaza, E., (eds) Springer Verlag, pp413-424.

Veloso, M., (1994) Planning and Learning by Analogical Reasoning, Springer Verlag, Berlin Heidelberg New York.

Watson I., (1998) Is CBR a Technology or a Methodology?, *Proceedings of IEA-98-AIE, Lecture Notes in Computer Science*, Springer-Verlag, Berlin Heidelberg New York.

Wilke, W., Bergmann, R., (1998) Techniques and Knowledge Used for Adaptation During Case-Based Problem Solving, *Proceedings of IEA-98-AIE, Lecture Notes in Computer Science*, Springer-Verlag, Berlin Heidelberg New York.