Retrieval Issues in Real-World CBR Applications

How far can we go with discrimination-nets?

Pádraig Cunningham*, Barry Smyth**
Donal Finn, Eamonn Cahill****

* Department of Computer Science
Trinity College
College Green
Dublin 2
Ireland
(+353-1-772941)
cnnnghmp@cs.tcd.ie

** Hitachi Dublin Laboratory
Trinity College
College Green
Dublin 2
Ireland
(+353-1-6798911)
bsmyth@vax1.tcd.ie
dfinn@vax1.tcd.ie
ecahill@vax1.tcd.ie

Abstract

In this paper we present a proposition and ponder a question. We propose that a useful perspective on analogical reasoning and CBR is to consider them on a continuum of abstraction of remindings. This is an alternative to the conventional view where CBR and analogical reasoning are seen as separate endeavours with analogical reasoning dealing with remindings between domains and CBR concerned with remindings within one domain. The question is how far towards the abstract end of the continuum can the index-based retrieval techniques that are effective in CBR be used (eg. discrimination networks). We are considering episode retrieval as a two stage process; the first stage being the initial filtering of the case base, and the second stage selecting the best case from this candidate set. We focus on the base filtering stage and conclude that discrimination networks are adequate for quite complex applications. However, problems arise when the system is required to support remindings at different levels of abstraction.

1 Introduction

Case-based reasoning (CBR) and analogical reasoning (AR) both retrieve episodes from memory based on the similarity of that episode to a case or scenario under consideration. Conventionally analogical reasoning is understood to be concerned with inter-domain remindings where the domain of the analog may be different to the domain of the target scenario. By contrast CBR is considered to depend on single domain remindings where the base and target cases come from the same domain. Our assertion is that this demarcation is artificial and the distinction is better characterised as a continuum of abstraction of remindings with CBR towards the concrete end and AR near the more abstract end. This assertion is based on the observation that CBR systems may be required to support remindings between different sub-domains of a 'single' problem domain.

In most CBR and AR systems case retrieval is a two stage process. First there is a pre-selection stage, often called **base filtering**, where a small set of candidate cases is selected. Then there is a **mapping** of the target case to these candidates to find which offers the best match. In straightforward CBR systems discrimination networks (D-Nets) are an effective method for organising the case memory to support base filtering. Towards the AR end of the spectrum, where remindings are more abstract, or where the system may need to support remindings at different levels of abstraction, it becomes more difficult to organise the case-base as a D-Net. The argument sometimes presented is that this dependence on indices will preclude remote remindings [Waltz '89] [Thagard '89]. In this paper we will look at some practical examples of CBR and assess the extent to which this is true.

We will begin by examining approaches to memory organisation and retrieval in AR and CBR. Some classical analogies will be discussed with emphasis on

the difficulties involved in triggering the remindings. We will present D-Nets as a typical example of an abstraction hierarchy and also D-Nets with redundancy which overcome some of the shortcomings of D-Nets. These ideas will be elaborated with a description of a typical CBR application that is adequately implemented as a redundant D-Net. Then we will look at a more complex CBR application in software design, requiring more abstract remindings, that causes problems for memory organisation based on indices. We will consider the use of more abstract indices and also index transformation as solutions to these problems. The last section reflects on the extent to which these modifications to the basic D-Net idea compromise its computational tractability and ease of setup.

2 Episode-Based Reasoning

Both analogical reasoning and case-based reasoning methods fall under the general category of episode-based reasoning (EBR) where problem solving knowledge is characterised as a set of episodes each representing the solution to a specific problem situation. A new problem (the *target*) is solved by retrieving a similar episode (*base episode*) from memory, and its solution is then modified to conform with the target situation. Clearly, the retrieval of an appropriate case is vital to the success of such techniques. Retrieval constitutes a massive search problem which is exacerbated by the fact that we are not concerned with complete matches but partial matches. Conventional methods take a two-stage approach. The initial retrieval stage (called *base filtering*) is responsible for selecting a small number of candidate episodes which are considered contextually similar to the target situation. The second stage (*mapping*) performs a detailed mapping between the target situation and each candidate episode to determine a single *best* episode for modification. The motivation for this two-stage approach is that the computational expense associated with the second stage is lessened because only a small number of candidates are

considered for mapping. Before discussing base filtering in more detail we will compare CBR and AR in the context of retrieval.

2.1 A Perspective on Retrieval

Most work on analogy has concentrated on inter-domain remindings where the relationship between the base episode and the analogous new episode is of an abstract or thematic nature (for example [Keane '87]). Alternatively, CBR research has focused on single-domain remindings where significant overlap of surface features exists between the base and target episodes. Such differences in the nature of remindings impose different constraints on the organisation of the episode memory and the retrieval process.

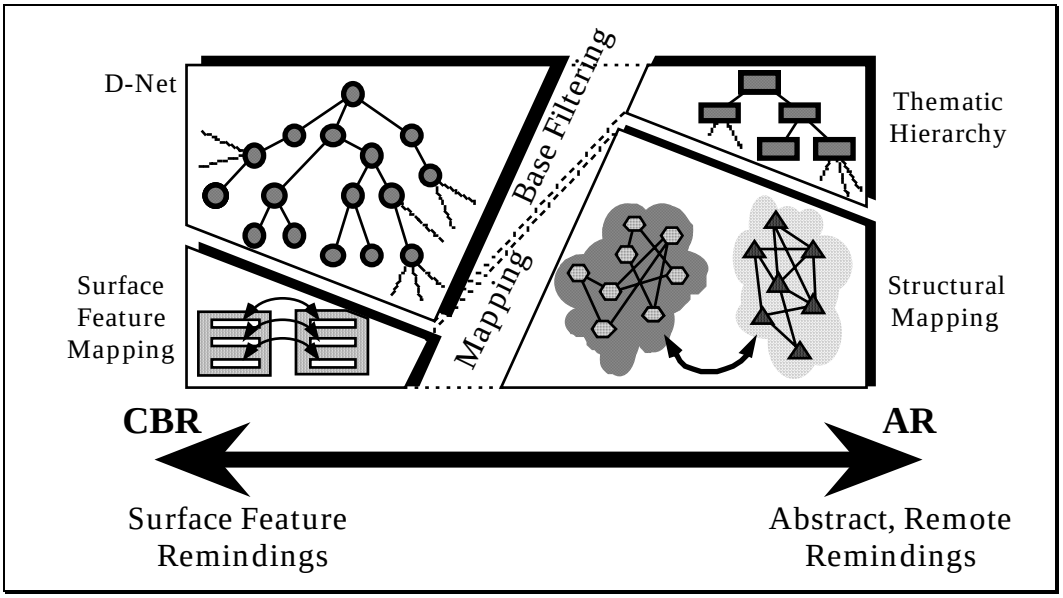

Figure 1. A perspective on retrieval in CBR and AR

In particular, as can be seen from Figure 1, much of the variation between AR and CBR is identified by a relative shift in emphasis from base filtering (in the case of CBR) to mapping (in the case of AR). Essentially, the surface feature based remindings of CBR necessitate simple, shallow mappings whereas the abstract remindings of AR require more complex 'structure mapping' type techniques (see [Falkenhainer '89]).

We are concerned with the organisation of episode memory from the point of view of base filtering. Episodes must be described in terms of their salient features. These descriptions should be structured not only to allow for efficient retrieval but also to facilitate remindings at the appropriate level of abstraction. These issues of indexing and remindings are discussed in the following section.

2.2 Memory Organisation

The organisation of episode memory must serve two objectives in base filtering; (a) to provide for adequate *remindings* so that all suitable episodes are considered and (b) to ensure for efficient *indexing* so as to guarantee fast retrieval of cases. Considering these in more detail:

Indexing

In general in EBR the expedient view is that cases should be indexed in order to support directed search during base filtering. However, it is evident that indexed memory will present difficulties in supporting remote or abstract remindings. The work of Waltz and Stanfill and that of Thagard and Holyoak is explicitly directed at cross domain remindings and memory organisation that supports abstract remindings, [Thagard '89 '90], [Waltz '89] [Stanfill '86]. However as a consequence, retrieval is only possible when cases are stored without indexing. This requires that memory is content-addressable in that all information about stored cases is matched with the target case and the 'best' match is returned. It is assumed that the exhaustive search implied in this approach is made feasible by parallel hardware.

Remoteness of Reminding

The other important characterisation of memory organisation is the remoteness of the remindings that are supported. The simplest perspective here is that CBR is concerned with reasoning within one domain where base filtering is done on the basis of surface features. AR involves inter domain remindings where

episodes share a structural rather than a semantic similarity and base filtering is based on abstract remindings.

The key issue concerning the role of indexing and reminding in memory organisation is that indexing cases according to their features permits the case base to be organised into an abstraction hierarchy thereby facilitating base filtering. However, abstraction and classification of these features becomes increasingly difficult as the nature of the remindings become more abstract. This issue is best considered if we examine two examples taken for the opposite ends of the EBR spectrum. The first example considers a purely structural analogy between electrical and mechanical systems, where it is difficult to uncover even the most abstract features that the two cases share, so the analogy is valid only on the basis of structural isomorphism between the cases. The second example examines a CBR system for estimating house prices that needs to support only the most superficial type of remindings.

Abstract Remindings: an AR example

Comparison of electrical circuits and mechanical systems provides a rich supply of engineering analogies. Figure 2 illustrates two very different systems that bear a strong structural similarity. Both exhibit damped oscillation as shown in the behavioural graph in the centre of the diagram. If the mass in the mechanical system is displaced it will oscillate about its equilibrium position with constant frequency and decreasing amplitude. If the capacitor in the electrical circuit is discharged its charge q will oscillate and decay to zero.

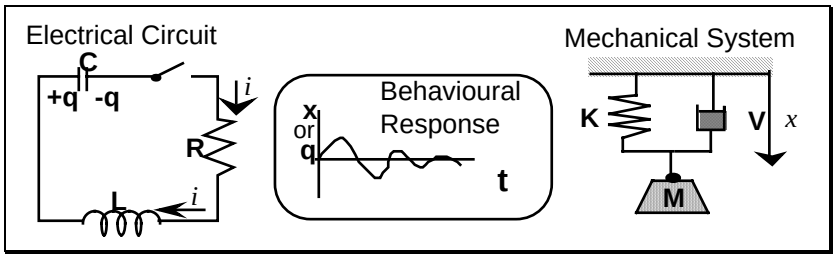

Figure 2. Two systems that exhibit damped oscillation.

This is quite a fundamental analogy in that both systems are governed by the same differential equation:

$$L\frac{d^2q}{dt^2} + R\frac{dq}{dt} + \frac{1}{C}q = 0 \quad \text{or} \quad M\frac{d^2x}{dt^2} + V\frac{dx}{dt} + K x = 0$$

This mathematical relationship can be expressed as a set of causal relationships that could be the basis for a structural mapping. However, what is of interest for us in this example is how one of these cases might be retrieved in base filtering as a potential analogue for the other. These cases share no surface features in common so the question is how could we plausibly index these with functional attributes that will capture the similarity? An abstract feature **damped-oscillation** captures the commonality very well. Even if the mechanical example were indexed as **damped-harmonic-motion** we could argue that this would be retrieved as a specialisation of **damped-oscillation.** The analogy would be more perspicuous if the capacitor and spring belonged to an abstract class **energy-reservoir** and the resistor and dash-pot shared an **energy-dissipater** superclass. The issue is, are we stretching credulity by expecting a knowledge-base to have these classifications?

Surface Remindings in a CBR example

Figure 3 shows two example cases from a case-based system called Rachman that can predict the selling value of a house given some details about it. The system contains a large case base of houses and their selling prices and it will retrieve a case or a set of cases describing similar houses and their selling prices. These prices can be adjusted depending on differences between the target and base cases to estimate the price of the target house. This system is comparatively straightforward but is equivalent to a host of potential CBR applications, for example loan risk assessment and help-desk assistants. The complexity of this problem is greatly relieved by having a well populated case base, so good matches can be found and the required adaptation is not difficult.

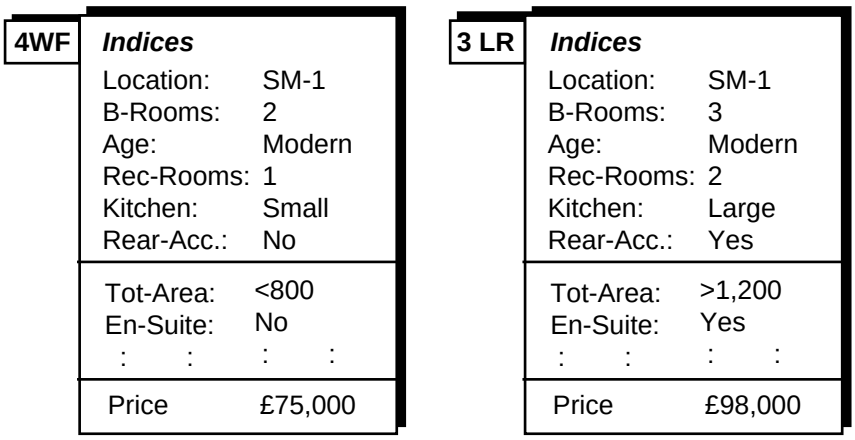

Figure 3. Two sample cases from the Rachman Case-Base

The cases are divided into two sets of features, the index features and the internal features. The index features are the most strongly predictive features and form the basis for the D-Net. The main problem with the D-Net approach is that it forces a strict ordering of the index features, in this example the cases might be organised first under location, then number of bedrooms, etc. However, different users may have different priorities; some, for instance, might consider the number of bedrooms to be more important than location. In addition, it will not be possible to retrieve matches for cases that have missing features as the retrieval process will not be able to search below the level of that feature in the network.

These problems are largely solved by introducing redundancy into the D-Net. This means that the network supports alternative orderings on the index features (see Figure 4). The extent to which redundancy can be introduced into the network is limited because the size of the network grows in proportion to the number of orderings supported. Retrieval in Rachman returns clusters of cases and the cases are ranked according to their frequency of occurrence in these clusters.

The purpose of this example is to show the success of index based retrieval and to illustrate some of the characteristics of an indexed case base. The success of this approach depends on having a case base that can be characterised by a

small set of indices that can be determined in advance. It is also important that the case base is well populated so that near or exact matches can be found.

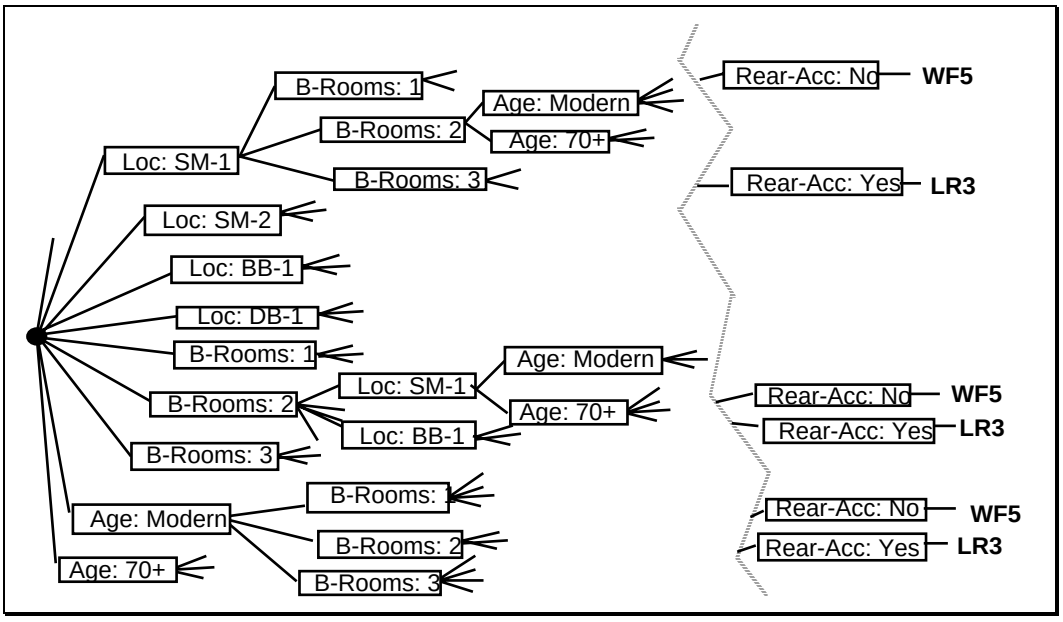

Figure 4. A portion of the Rachman Case-Base organised as a D-Net with some redundancy.

2.3 A Scheme for Achieving Remote Remindings in a CBR Structure

It is evident from the two examples in the previous section that a strong demarcation is thought to exist between AR and CBR. Fundamentally, the common perception is that in inter-domain AR situations, indexing using abstract remindings is problematic and therefore exhaustive search with subsequent structural mapping is the only feasible approach [Thagard '90] [Waltz '89]. At the other end of the spectrum, the accepted wisdom for CBR systems, is that indexing should be based on obvious surface features thereby permitting efficient base retrieval using an abstraction hierarchy such as a D-Net.

Our experience, particularly in research on software design using CBR, is that this Single-Domain v's Inter-Domain division is artificial in that no strict demarcation exists. In particular, systems designed to operate within 'one' domain may be required to support mappings between sub-domains of that

domain. Rather, what we argue is that a continuum of abstraction with CBR at one end and AR at the other can be thought to exist. Between these two extremes lies the complete spectrum of remindings at different levels of abstraction.

In order to support remindings across increasingly remote domains and at the same time to reflect the fuzzy nature of the degree of semantic commonality of cases, a hierarchical index structure is employed for each case as illustrated in Figure 5.

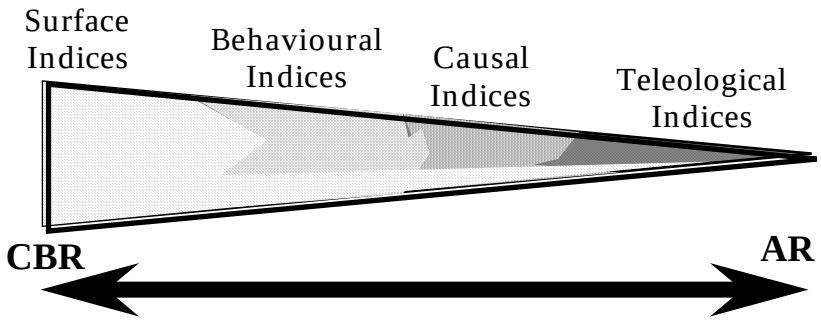

Figure 5. A Perspective on Indexing

As we progress up the hierarchy there is a transition from data-type indices to more knowledge-type indices. This scheme attempts to characterise the different degrees of knowledge humans can use to reason about a case and would appear to be more psychologically plausible than a flat index structure. We will explain these indices in the context of the example of the LCR circuit and mechanical spring-dash pot system presented in Section 2.2.

Surface Indices reflect observable features in a case (capacitor, coil and resistor *etc.*) and would be the set of indices used at the strictly CBR end of the spectrum.

Behavioural Indices embody emergent behaviour of the case stemming from the interaction of the components within the case. They capture the characteristics of the overall system (e.g. damped, oscillatory motion) and it

is at this level that matching between the electrical and mechanical systems could be achieved.

Causal Indices encode explanatory knowledge about how the behaviour comes about as a consequence of the interaction of the sub-entities within the case e.g. the inertial effects of the mass (coil) transfers energy between the two energy reservoirs, namely the spring and the mass (capacitor and coil). During the transfer, energy is dissipated by the dash-pot (resistor).

Teleological Indices detail the purpose or goal of the case (to form a tuned radio circuit or to act as a car suspension for example).

This example of AR demonstrates the conceptual and theoretical merits of the proposed indexing scheme for cross-domain remindings. In the next section we will report on the practical issue of its computational tractability for a real-world CBR problem.

3 Complex Remindings in Déjà Vu

The Déjà Vu problem domain has already been introduced in [Smyth '92] and so will only be described in outline here. Déjà Vu is a CBR system for design of plant control software; an example of the type of code is shown in the Solution section in Figure 6. The software is for controlling loading and unloading equipment in a steel mill. The code is expressed in this network representation that is compilable into executable code. This sample case controls the movement of a buggy carrying an empty spool. Buggy*1 is a two speed buggy, so stopping is a two stage process with the buggy switching to its slower speed 200mm from its destination. This case is a sub-component of a complete solution sequence.

Describing the scenario in more detail; this buggy is part of a system for carrying coils of steel or empty spools between a storage area (called a skid)

and a tension reel where it is mounted on the rolling mill. The load is not mounted directly on the buggy but is carried on a lifting device that is mounted on the buggy. This lifter is used to adjust the height of the load for loading and unloading. The lifter can be a one or two speed device - a component case for a two speed lifter is shown in Figure 7. It can be seen that the solution for the two speed lifter has the same structure as the buggy case described above. This is problematic because these two cases do not share important surface features. So if this lifter case were a target case, the useful Advance*B1*Spool case would not be retrieved using this indexing scheme.

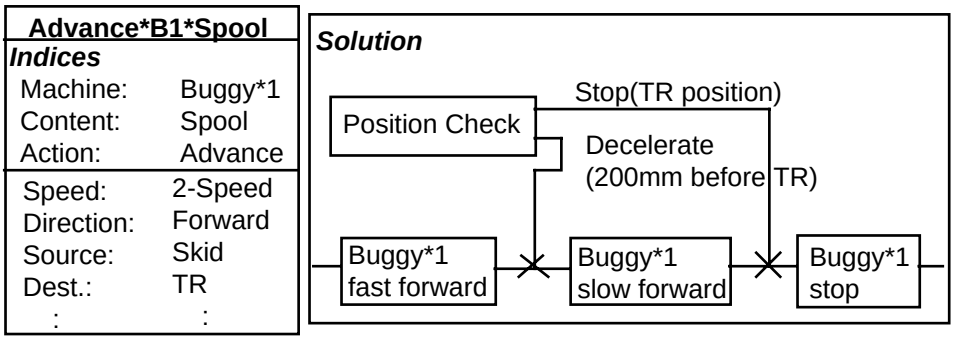

Figure 6. An example case from Déjà Vu.

Building a D-Net to characterise a simple domain such as RACHMAN's property domain is a fairly straight forward task. Such a domain can be modelled in terms of its components (an identifiable, finite set), and thus can be completely described. However in more complex domains concerned with reasoning about actions and change this becomes a far more difficult task requiring consideration of more complex behavioural and functional domain features.

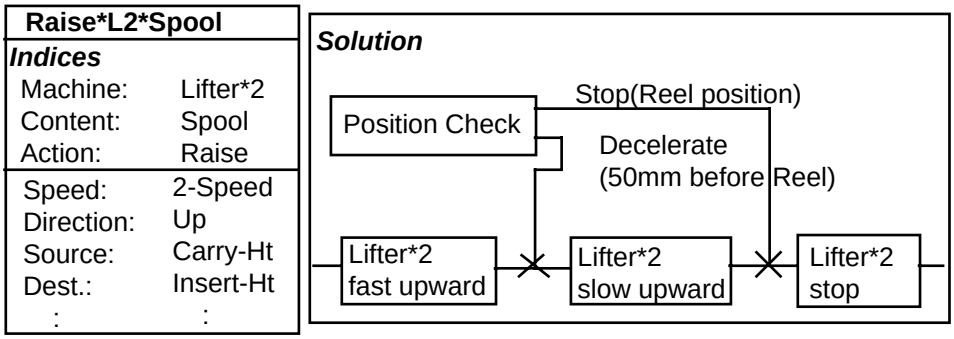

Figure 7. A two-speed lifter case.

The current example from Deja Vu has attempted to describe two cases in terms of surface features, indicating the ACTION, VEHICLE, and CONTENT of the cases. The problem is that the observed difference in action types causes the cases to be indexed under different routes within the case-base, even though behaviourally and structurally the cases are very similar. There are two approaches to solving this problem, effectively drawing the cases closer together: abstract indexing, and index transformation.*

3.1 Abstract Indices

The problem that we are addressing concerns the fact that even though the two cases differ significantly in terms of their surface features (BUGGY*1 and ADVANCE against LIFTER*1 and RAISE), they exhibit strong behavioural and structural similarities. These similarities are not captured by the chosen indices and so the cases are distant from each other within the case-base.

One solution is to capture this behavioural and structural similarity by adding abstract behavioural indices, such as BEHAVIOUR = MOTION and BEHAVIOUR-TYPE = 2-SPEED. Now the two cases appear as siblings within their appropriate behavioural route. This is illustrated below in Figure 8 where the above behavioural indices are used to capture the inherent similarity between the ADVANCE and RAISE cases. Therefore, in this 'behaviour' section of the net these cases are classified as similar without the need for index processing techniques such as index transformation.

Referring back to the pyramid of index types and the CBR-AR continuum shown in Figure 5, as more abstract remindings are required, so these must be represented as different types of routes within the case-base structure. In this way cases are considered for retrieval at different levels of abstraction. In the

* It is worth noting that if there were existing two speed lifter cases in the case base then this problem would not arise as they would offer a better match. However, the essential point that there is a similarity that is not being captured continues to be valid.

Déjà Vu example we are concerned with two levels, namely component and behavioural, and it is at the behavioural level that the inherent similarities between the cases becomes apparent.

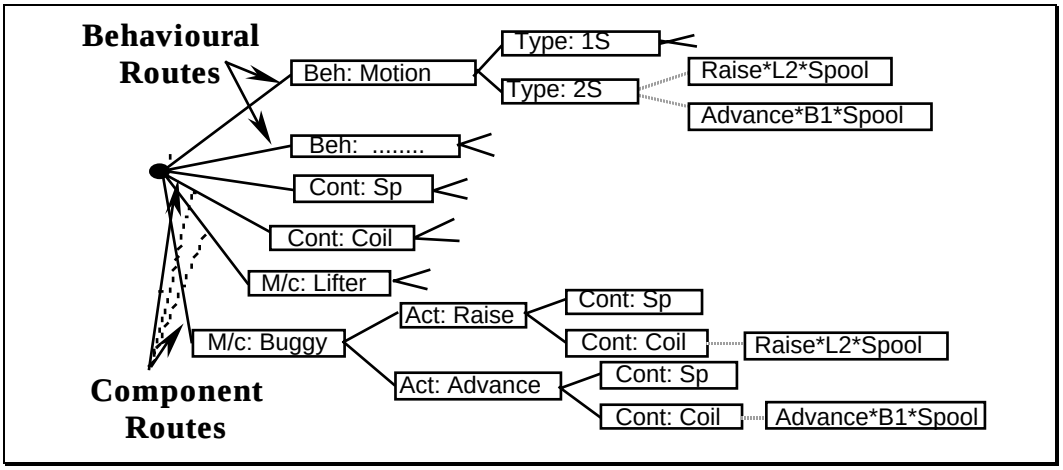

Figure 8. Incorporating behavioural indices facilitates recognition of behavioural similarity.

Introducing these behvioural features does solve the problem but there are some caveats. This 'closeness' in the network depends on the redundancy of the net supporting just the right ordering of the index features to bring the cases together. In addition 'less relevant' matches based on surface features will also be retrieved by the base filtering. This forces a consideration of what exactly are the requirements on base filtering. One view would be that the onus is on the base filtering process to only produce cases that are truly relevant. This appears to require that the relative importance of indices be context dependant - that dynamic indexing be supported. Introducing redundancy into the D-Net does support different index orderings but it cannot *suppress* orderings that are not relevant in particular contexts.

The stance taken in Déjà Vu is that it is acceptable for the base filtering to retrieve cases that may subsequently prove to be irrelevant: the important point being that the set of cases returned by the base filtering must contain the best case. In the case mapping phase, these cases are examined to determine the one

that can most easily be adapted to fit the target problem (see [Smyth '92b] for more on adaptation driven case selection).

3.2 Index Transformation

Quoting from Sycara and Navinchandra:- "Index transformation changes given salient features to match the indices under which previous cases have been stored making previously inaccessible cases accessible." [Sycara '91] In the current situation this involves altering one or more of the indices so it is evident that a strong model of the problem domain is required to support transformation. In Déjà Vu the domain is modelled using a frame representation that captures the attributes of the domain concepts and the interactions between them. In this model Buggy and Lifter are sub-classes of Mover and inherit the Speed slot from that class. The appropriate transformation in this situation involves relaxing the Buggy index and seeking cases indexed on siblings of Buggy. In the current system, on retrieval failure, there is no way of knowing in advance which indexes need to be relaxed so all indexes need to be transformed in turn. This greatly increases the retrieval time. This approach has the advantage that it escapes the rigidity of the static indexing by exploiting the available domain model.

In the absence of a close match the system is retrieving a case that is structurally similar but has different surface features to the target case. This means that the adaptation task is more substantial. The very pragmatic argument could be presented that there should be little interest in more abstract remindings in Déjà Vu because the adaptation process would be too complex.

4. D-Nets: *How far can they go ?*

So we have been discussing D-Nets as a means of organising an indexed case-base in order to facilitate base filtering. The main advantages of this approach are that; the case-base is easy to set up as little knowledge engineering is

required, and case retrieval in base filtering is computationally tractable. If we consider CBR and AR as a continuum of abstractions of remindings and recognise that D-Net based retrieval works for simple CBR problems based on surface features then:- 'how far toward the *abstract* end of this continuum can D-Nets be used?'.

Two problems that need to be addressed in attempting to develop CBR systems more sophisticated than the basic Rachman system described section 2.2 are:-

- **Dynamic indexing:** the need for more flexibility in the prioritizing indices.
- **Abstract remindings:** the need to recognise near if not exact matches.

Dynamic Indexing

The hierarchy in a D-Net reflects an implicit prioritization of indices. In the Rachman example the D-Net supports a few different indexing orderings in that cases can be retrieved giving 'location' priority over 'number-of-bedrooms' or vice versa. It is important to emphasise that the number of routes in the net increases directly with the number of different orderings supported so this measure of introducing redundancy must be used sparingly. This goes some way towards addressing the problem but it is not dynamic indexing, it is a selection of conflicting static indexes. This issue arose in the discussion of the Déjà Vu system in section 3 where the desired reminding was on the basis of a feature more abstract than the surface features and in that situation remindings based on the surface features needed to be suppressed. As stated in section 3.1, this problem is solved in Déjà Vu by accepting 'false' remindings from the base filtering and laying the onus on the case mapping phase to select the best candidate case. We would consider that this solution is being stretched to the limit in handling the two classes of index in Déjà Vu and would not be adequate for a system meant to support broader classes of remindings.

Abstract remindings

These attempts to overcome the problems associated with static indexing compromise the computational tractability of the D-Net as the number of routes in the net increases directly with the number of orderings supported. In the same manner, the need to support abstract remindings compromises the ease of setup of the case-base: the ability to recognise abstract matches requires that the system have a comprehensive domain model. Because of this, the argument that CBR requires less knowledge engineering than other expert system techniques becomes less valid.

Pragmatically speaking, the task of CBR systems (and indeed, expert systems) is to provide a facility for automatic problem solving within a given application domain. Of primary importance to CBR is the characterisation of that domain through a representative set of cases. Organising these cases in terms of a D-Net affords an efficient base-filtering process, however the effectiveness of the network in capturing the domain is solely dependant on the choice of indices. In choosing a set of indices we are providing a mapping from the complete domain model (which is in general too complex to capture using conventional knowledge-base methods) to the case-base structure (index space) being used to approximate this model (see Figure 9).

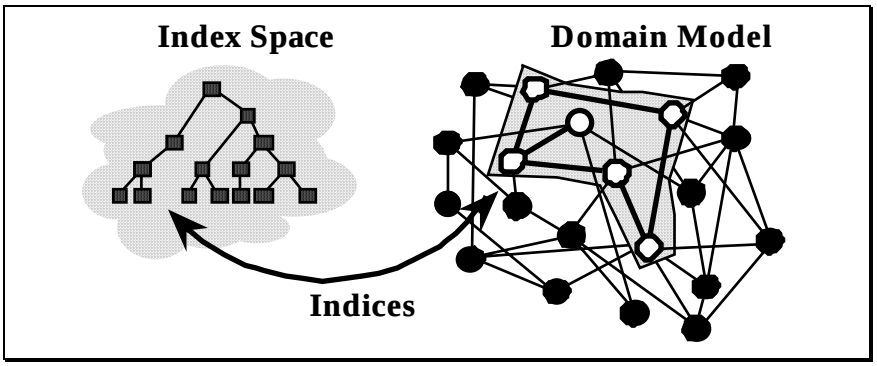

Figure 9. Mapping between the domain model and the case-base.

The selected indices determine the set of problems that can be solved by the system. Using conventional CBR methods, the complete set of solvable

problems is exactly those that can be described in terms of the chosen indices. Of course this assumes that the adaptation process can transform any given close match into the appropriate solution. This is clearly a more tractable prospect in a system with a well populated case-base, since for any given (solvable) situation the probability of locating a case which is similar enough to be correctly adapted, is higher than in a similar system with a sparse case-base. Sparse case-bases require more complex adaptation methods capable of adapting between more distant problems, retrieved on the basis of more abstract remindings.

Situations falling outside of the coverage of the index space require a restructuring of the case-base to incorporate the additional indices necessary to describe such situations. Handling a problem situation which is not covered by the index space requires that a mapping be set up between the available indices and the more abstract domain model relations so that the necessary indices may be abstracted out and used to describe and solve the new problem.

Abstract Indices and D-Nets

We have seen in Section 3 that even within what is ostensibly a single-domain problem (automatic programming for plant control), the need for cross-domain remindings arises. Surface features alone were found to be inadequate for generating the required retrievals. Of the two approaches considered as a means of addressing this problem, *Index Transformation* and *Abstract Indices*, neither was found to provide an ideal solution. Nevertheless the problem did prove to be tractable when two levels of abstract indices were interleaved in a D-Net.

The drawback of mixing multi-level indices within the D-Net is that it clearly exacerbates the problems associated with the net; primarily the explosion in the number of possible orderings. Therefore we are limited as to how far we can feasibly proceed with the index hierarchy with current base filtering techniques.

This problem would be alleviated somewhat if the base filtering proceeds solely on the basis of indices from the same level of Figure 5. This would enable the use of a disjoint net for each level of the index hierarchy. The construction of the net(s) would then be less complicated, and case retrieval within a net much less computationally expensive.

In summary, although the index hierarchy proposal is conceptually attractive and a reduced version of it computationally tractable, further research is needed to investigate suitable mechanisms for base filtering with such sophisticated indexing.

5 Conclusion

Rather than consider AR and CBR as two different tasks defined as inter-domain and single-domain respectively we propose that they be considered as a continuum of abstractions reflecting the level of abstractness of the reminding required. This view is motivated in part by the observation that potential CBR applications can contain sub-domains, between which remindings should be supported.

Using this continuum as a basis, the question considered is; how far along this continuum can one use the D-Net techniques that are so useful in simple CBR applications. The conclusion is that once D-Nets are adapted to support remindings at different levels of abstraction the advantages of ease of setup and computational tractability are compromised. We have argued that the Déjà Vu system is able to support remindings based on surface and behavioural features but that this is about the limit. A D-Net based system supporting broader classes of remindings would be difficult if not impossible to maintain.

References

Falkenhainer B., Forbus K.D., Gentner D., (1989) "The Structure Mapping Engine: Algorithm and Examples", *Artificial Intelligence*, Vol.41, pp1-63.

Keane M., (1987), "On Retrieving Analogues When Solving Problems", *The Quarterly Journal of Experimental Psychology*, Vol. 39A, pp29-41, 1987.

Smyth B., Cunningham P., (1992), "Déjà Vu: A Hierarchical Case-Based Reasoning System for Software Design", in *Proceedings of 10th. European Conference on Artificial Intelligence,* Vienna, Austria, ed. Bernd Neumann, Wiley & Son, pp587-589, 1992.

Smyth B., Keane M.T., (1993), "Retrieving Adaptable Cases: The Role of Adaptation knowledge in Case Retrieval", submitted to IJCAI '93.

Stanfill C., Waltz D., (1986), "Toward memory-based reasoning", *Communications of the ACM,* pp1213-1228, Vol. 29, No. 12, December 1986.

Sycara K.P., Navinchandra D., (1991) "Index transformation techniques for facilitating creative use of multiple cases", in *Workshop on Artificial Intelligence in Design at Twelfth International Joint Conference on Artificial Intelligence,* J.F. Gero, F. Sudweeks, eds., Sydney, Australia, 1991.

Thagard P., Holyoak K.J., (1989), "Why indexing is the wrong way to think about analog retrieval". in *Proceedings of DARPA Case-Based Reasoning Workshop 1989*, Morgan Kaufmann, San Mateo, California, pp36-40, 1989.

Thagard P., Holyoak K.J., Nelson G., Gochfeld D., (1990), "Analog Retrieval by Constraint Satisfaction" in *Artificial Intelligence*, Vol. 46, pp259-310, 1990.

Waltz D.L., (1989), "Is indexing used for Retrieval". in *Proceedings of DARPA Case-Based Reasoning Workshop 1989*, Morgan Kaufmann, San Mateo, California, pp41-44, 1989.