EXPERIMENTS ON ADAPTATION-GUIDED RETRIEVAL IN CASE-BASED DESIGN

Barry Smyth
Hitachi Dublin Laboratory,
Trinity College Dublin,
Dublin, IRELAND.
Phone: +353-1-6798911
Fax: +353-1-6798926
EMail: barry.smyth@hdl.ie

Mark T. Keane
Department of Computer Science,
Trinity College Dublin,
Dublin, IRELAND.
Phone: +353-1-7021534
EMail: mark.keane@cs.tcd.ie

ABSTRACT

Case-based reasoning (CBR) has been applied with some success to complex planning and design tasks. In such systems, the best case is retrieved and adapted to solve a particular target problem. In general, the best case is that which can be most easily adapted to the target problem (as the overhead in adaptation is often very high). Standard CBR systems use semantic-similarity to retrieve cases, on the assumption that the most similar case is the best or easiest case to adapt. However, this assumption can be shown to be unwarranted. In this paper, we report a novel retrieval method, called *adaptation-guided retrieval*, that is sensitive to the ease-of-adaptation of cases. In the context of a CBR system for software-design, called Déjà Vu, we show through a series of experiments that adaptation-guided retrieval is more accurate than standard retrieval techniques, that it scales well to large case-bases and that it results in more efficient overall problem-solving performance. The implications of this method and these results are discussed.

Keywords: *Case-Based Reasoning, Retrieval, Adaptation, Software Design.*

Declaration:

This paper has not already been accepted by and is not currently under review for a journal or another conference. Nor will it be submitted for such during IJCAI's review period.

EXPERIMENTS ON ADAPTATION-GUIDED RETRIEVAL IN CASE-BASED DESIGN

ABSTRACT

Case-based reasoning (CBR) has been applied with some success to complex planning and design tasks. In such systems, the best case is retrieved and adapted to solve a particular target problem. In general, the best case is that which can be most easily adapted to the target problem (as the overhead in adaptation is often very high). Standard CBR systems use semantic-similarity to retrieve cases, on the assumption that the most similar case is the best or easiest case to adapt. However, this assumption can be shown to be unwarranted. In this paper, we report a novel retrieval method, called *adaptation-guided retrieval*, that is sensitive to the ease-of-adaptation of cases. In the context of a CBR system for software-design, called Déjà Vu, we show through a series of experiments that adaptation-guided retrieval is more accurate than standard retrieval techniques, that it scales well to large case-bases and that it results in more efficient overall problem-solving performance. The implications of this method and these results are discussed.

1 Introduction

Most case-based reasoning (CBR) systems replace a first-principles problem-solver with cases and knowledge-weak adaptation rules to modify these cases. The success of such systems depends on selecting the best possible case during retrieval. The majority of CBR systems retrieve cases using semantic-similarity metrics; the assumption being that the most semantically-similar case to the target problem will be the most useful and easiest to adapt. However, this assumption is not always warranted; the most similar case may *not* be the easiest to adapt and may even be impossible to adapt.

This realisation has led some researchers to augment semantic similarity with other factors. Kolodner (1989) proposed that some mappings between a target problem and a candidate case should be preferred over others if they were, for example, more *specific* or *goal-directed*. She also argued that "easily-adapted" matches should be preferred over "hard-to-adapt" matches. Goel's (1989) KRITIK system also prefers candidate design-cases that satisfy the functional specifications of the target design and hence have easily-adaptable matches. Birnbaum, Collins, Brand, Freed, Krulwich & Pryor (1988) proposed a system that learns to index cases on the basis of their adaptability, overriding semantic similarity where appropriate.

Their system avoids cases with feature combinations that were difficult to adapt in previous problem-solving episodes.

We agree with the spirit of these proposals but favour a different solution. All of the above systems involve an across-the-board promotion (or demotion) of certain matches based on their *likely* rather than their *actual* ease-of-adaptation. They make an "educated guess" as to the adaptability of cases rather than a detailed assessment of their adaptation requirements. We advance a novel technique, called *adaptation-guided retrieval* (AGR), that assesses the adaptation requirements of cases during retrieval. AGR makes direct use of specially-formulated, adaptation knowledge to determine simple surface-changes, structural transformations, and complex interactions (see Smyth & Keane, 1994) [Our integration of adaptation knowledge into retrieval is influenced by techniques used to introduce other forms of domain knowledge into retrieval (see Cain, Pazanni & Silverstein, 1991; Veloso, 1992)]. Furthermore, AGR works without incurring the full cost of adaptation during retrieval. Indeed, AGR can be more efficient and effective than standard methods in CBR.

AGR is implemented in Déjà Vu, a case-based reasoning system for software design (see section 2). In section 3, we show how adaptation-guided retrieval works; how cases are selected based on their adaptation requirements and how subsequent adaptations are predicted during retrieval. In section 4, we present experimental evidence to demonstrate some of the performance and competence advantages offered by adaptation-guided retrieval.

2 Déjà Vu & The Plant-Control Domain

Déjà Vu is a case-based reasoning system for software design boasting two main novelties. Firstly, it uses adaptation-guided retrieval. Secondly, it integrates case-based and decompositional design methods by imposing a hierarchical structure on the case-base such that complex problems are represented as hierarchies of cases at varying levels of abstraction (as discussed in Smyth & Cunningham, 1992a, 1992b). The primary application domain of Déjà Vu is plant-control software design. Plant-control programs regulate the action of autonomous vehicles within real industrial environments. The examples in this paper are taken from a steel mill environment where a system of track-bound vehicles (called coil-cars) load and unload spools and coils of steel. Figure 1(a) illustrates a sample plant layout and 1(b) a schematic of a basic Load/Unload task with a coil-car, a mill (tension-reel), a loading-bay (skid), and a spool or coil of steel.

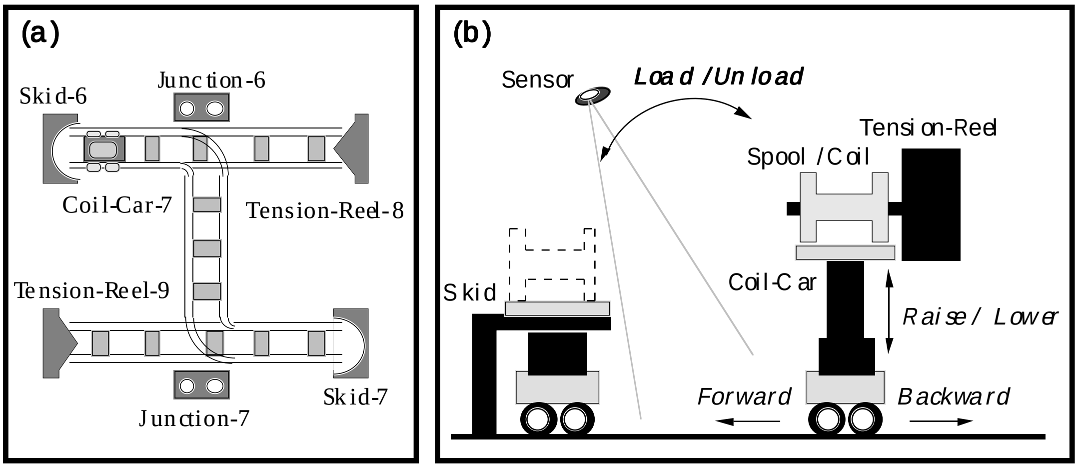

Figure 1. (a) Track Layout; (b) Load/Unload Task Schematic

Déjà Vu's decompositional design component enables complex problems to be broken up into simpler tasks by the retrieval of abstract cases. Actual solution code is then produced by the retrieval and adaptation of the appropriate design cases with the resulting solution segments being integrated into overall solution structure on the fly. Problem solving activity is co-ordinated using a blackboard architecture with dedicated knowledge agents handling such tasks as indexing, retrieval, adaptation, decomposition, and integration.

3 Adaptation-Guided Retrieval

Déjà Vu retrieves the best case by determining the adaptation requirements of candidate cases during retrieval. In this section, we outline Déjà Vu's adaptation component, how AGR works and present some examples of its use.

3.1 Déjà Vu's Adaptation Component

Déjà Vu's adaptation component adapts candidate cases using two distinct forms of knowledge: (i) *adaptation specialists* that perform specific, local modifications to cases, and (ii) *adaptation strategies*. that solve problematic interactions within cases.

Adaptation specialists correspond to packages of design transformation knowledge concerned with a specific adaptation task. Each specialist can make specific, local modifications to a retrieved case. For instance, in the plant-control domain, retrieved "move" cases often differ from a target problem in the speed of the coil-car (one- or two-speed). So, Déjà Vu has a dedicated *speed specialist* to modify the coil-car speed in retrieved cases to meet the appropriate target specifications (see Figure 2). Specialists contain two parts: (i) *capability* - information describing the nature of its adaptation task (e.g., the specialist in Figure 2 is designed to alter the speed constraint of a movement task from 1-speed to 2-speed). (ii) *action* - the procedural know-how needed to perform a particular kind of adaptation (e.g., to upgrade the speed of a case a number of additional solution nodes must be added to the 1-speed solution chart). In short, the capability information describes *what* must be adapted and the action information describes *how* this adaptation can be carried out. As we shall see, it is the capability information that allows specialists to be used during retrieval. During adaptation many specialists may act on the retrieved case to transform it into the desired target design. Thus, through specialist activity, the differences between the retrieved case and the target are reduced in a piecemeal fashion.

```
Capability
  (:TASKS       Move)
  (:MAPPINGS    ((VEHICLE CONSTRAINT-SPEED Target-Speed?)
                 (VEHICLE CONSTRAINT-SPEED Base-Speed?)))
  (:TESTS       (eq Target-Speed? 2-SPEED)
                (eq Base-Speed?   1-SPEED))

Action
  (INSERT-COMMAND
    (Def-Command Move <vehicle> Fast <direction>)
    :BEFORE
      (Def-Command Move <vehicle> Slow <direction>))
  (INSERT-COMMAND
    (Def-Command Distance-Check <vehicle>
                                <slowing-distance>
                                <orientation>
                                <destination-location>)
    :BEFORE
      (Def-Command Move <vehicle> Slow <direction>))
```

Figure 2. A Speed Specialist

Adaptation strategies deal with interactions that arise during the adaptation of a case by the specialists. Specialists are local and therefore ignorant of global interactions between case elements that may lead to problem-solving failures; interactions cause problems in many planning and automated design systems (see e.g., Hendler, Tate & Drummond, 1990). Déjà Vu's adaptation strategies detect and repair different classes of interactions that arise. The strategies are organised in terms of the interactions they resolve and each is indexed by a description of the type of failure it can repair. Each strategy also has a set of repair methods for fixing a particular interaction.

For example, one common interaction involves the effect of one event preventing the occurrence of a later event. Figure 3(a) depicts this situation; a goal event (1) is prevented by the disablement of one of its preconditions (2), the precondition having been blocked by some earlier event (3) causing a conflicting state (4). This *blocked-precondition interaction* could occur when the speed of a coil-car is increased (during adaptation), causing a fuel availability problem that results in the coil-car running out of fuel (fuel being a precondition of the movement goal). This interaction can be repaired by adding an event before the blocking event (3) that prevents its blocking effect; for example the taking on of more fuel before initiating the move. The blocked pre-condition adaptation strategy contains a description of this situation along with appropriate repair methods.

Another type of interaction, a *balance-interaction*, can occur when the value of one state is proportionally dependent on another (see Figure 3(b)). Here, some necessary goal-achieving event (1) has a precondition state (2) that depends on another state (3) that has resulted from some other event (4).

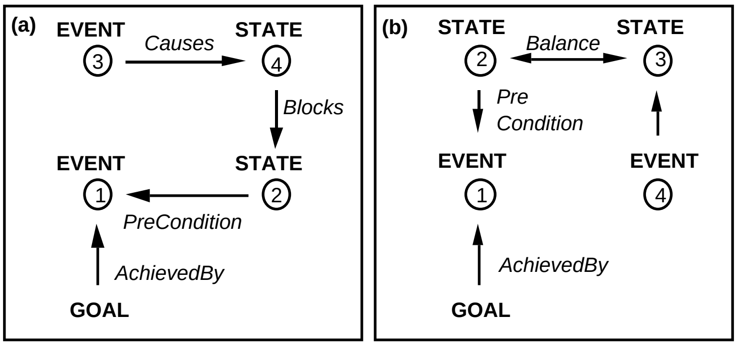

Figure 3. (a) Blocked-Precondition; (b) Balance-Interaction

For example, before moving a coil-car across the factory floor the height of the carrying platform must be adjusted to accommodate the load being transported; there is a balance condition between the height of the lifting platform and the diameter of the coil of steel being carried. If this balance is not properly maintained then a failure may occur (the coil-car may collide with an overhead obstacle).

The system currently uses 10 strategies to deal with all the interaction problems that arise in the plant-control domain. Our investigations suggest that these strategies are applicable to other domains, although others new ones may also be required. Hammond's (1989) CHEF uses similar types of knowledge to identify failures during meal planning (although they are not used for retrieval purposes).

3.2 The Adaptation-Guided Retrieval Procedure

Table 1 shows the three-stage process used to determine the adaptation requirements of cases during retrieval. *Candidate Selection* is a base-filtering stage that quickly eliminates irrelevant cases from further consideration. Basically, it removes any cases that have no specialists in common with the target specification. This stage treats all adaptable features as equally relevant and simply locates cases that are potentially adaptable to the target situation.

```
Input:     T,    a target specification; CB, a case-base
           AK,   adaptation knowledge

Output:    C,    the most "adaptable" case
           AK',  its relevant adaptation knowledge

1.  Candidate  Selection

     1.1   Locate candidate cases (CB') with features
           that can be adapted to those of the target.
     1.2   For each candidate, collect its specialists.

2.  Assessing  Local  Adaptability

     2.1   Compute  Case  Coverage --
            Map target and case features that are
            adaptable and remove any case that
            leaves some portion of the target
            unmapped (uncovered).
     2.2   Compute  Local  Adaptability --
            Estimate the complexity of each case's
            local adaptation requirements in terms
            of the relevant specialists.

3.  Assessing  Global  Adaptability

     3.1   Find  Applicable  Strategies --
            For each case collect any applicable
            strategies.
     3.2   Compute  Global  Adaptability --
            Estimate the complexity of each case's
            global adaptation requirements
            (possible interaction failures) in
            terms of the applicable strategies.
```

Table 1. The Adaptation-Guided Retrieval Procedure

In the *Assessment of Local Adaptability* the target's features are aligned (or mapped) with those of candidate cases. A feature mapping is only constructed if it is deemed adaptable, that is if there is a specialist to support the mapping. Briefly, a case is said to *cover* the target if some feature of the case can map on to each relevant feature of the target and if all of these mappings are adaptable. A local adaptability metric is applied to the remaining cases to estimate their ease of adaptation in terms of their applicable adaptation specialists.

Finally, during the *Assessment of Global Adaptability* the strategies that are applicable to each of the remaining candidates[1] are determined and a second metric is used to grade these cases according to the different repair methods that are suggested by each strategy. Different strategies are differentially weighted

[1]This is somewhat analogous to Veloso's (1992) interacting footprint similarity metric in that both attempt to address and assess the impact of possible interaction problems.

according to the amount of change their repair strategies incur. Some repair methods will significantly reorder a proposed solution whereas others may just require a simple deletion of an existing goal structure. Overall, the candidates are ordered according to both their local and global adaptability and the case that minimises both measures is chosen.

The output of the retrieval stage is a ordered set of candidate cases, their feature mappings, and the adaptation specialists and strategies applicable to each candidate. So, AGR is unlike conventional retrieval methods which simply return the chosen case, the feature mappings, and a similarity measure, with no support for adaptation and repair.

3.3 An Example

The following example works through a sample retrieval session taken from the plant-model of Figure 1(a). The target problem is to move coil-car-7 from tension-reel-9 to skid-7 using 2-speed motion carrying coil-1, a coil of steel and the case memory contains just a single case for moving a coil-car from tension-reel-8 to skid-6 using 1-speed motion, and carrying no load. The adaptation knowledge consists of a number of specialists designed to cater for transformations involving speed, direction, start and end locations, and the contents of vehicles in movement tasks. Two strategies are relevant; the blocked-precondition strategy and the balance-interaction strategy, both mentioned above.

Figure 4 is a representation of the types of structures built during retrieval. Since the target and base differ in terms of speed, direction, locations, and vehicle content, a variety of relevant specialists are activated and shown. A number of points are worth noticing here. First, only *relevant features* (i.e., features that are adaptable) partake in the mapping process. This contrasts with many knowledge-weak retrieval methods that *have to* consider the mapping of all specified problem features[2]. As a result adaptation-guided retrieval computes significantly fewer mappings than other methods. Secondly, at this early stage non-adaptable cases can be identified and eliminated. For example, if a speed specialist did not exist then

---

[2]One could of course build in some notion of relevance or context but again traditional models tend to be overly specific, prohibiting some cases from consideration, or overly general thereby forcing unnecessary mappings and the consideration of non-adaptable (but similar) cases. AGR provides a model of relevance that is directly based on the adaptation knowledge not on additional similarity knowledge.

Déjà Vu would have no alternative but to find a different case, namely one that matched exactly on speed.

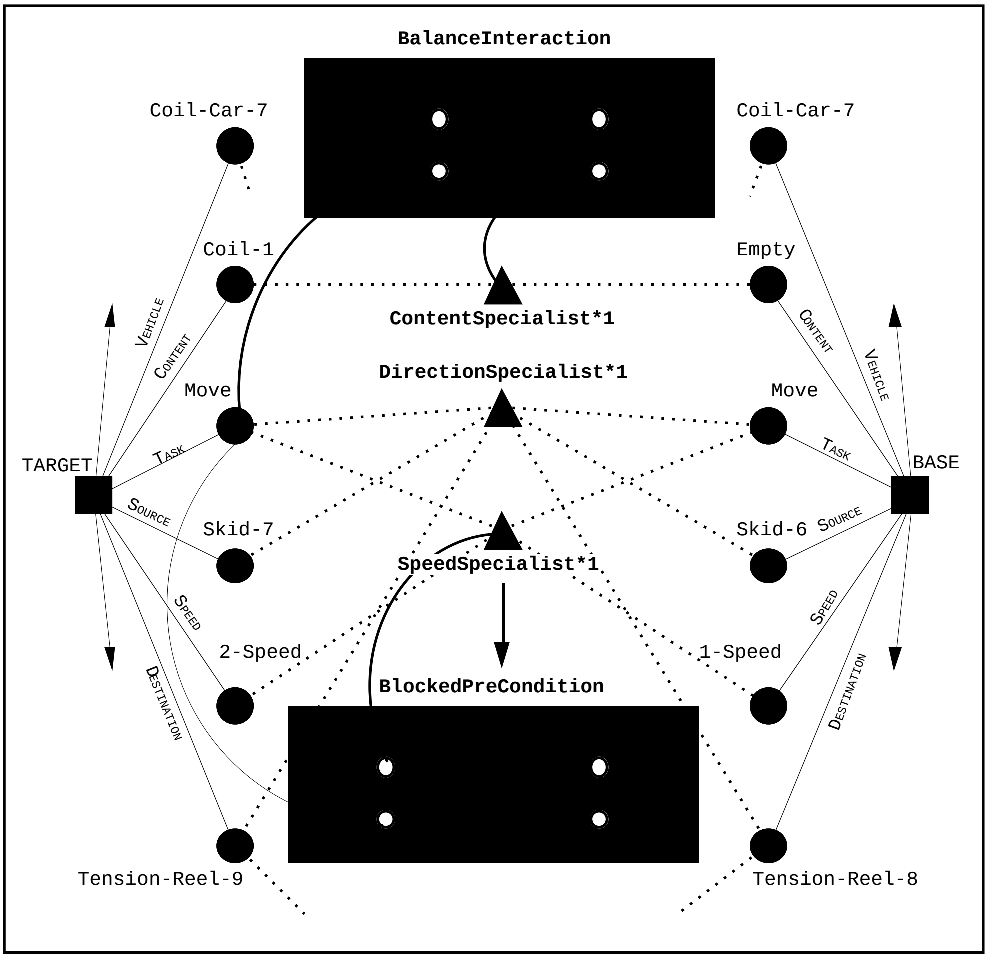

Figure 4. Retrieval Snapshot

In this example, when the specialists are applied (during adaptation) further problems in the form of blocked-preconditions and balance-interactions arise (see Figure 4). For instance, the content-specialist is set to change the base solution so that the target coil-car is carrying the target coil. However, there is a balance condition between the coil diameter and the carrying-height of the coil-car. This is detected at retrieval time because the content-specialist is known to affect a balance-state feature. Consequently, the balance-interaction strategy is stored with the applicable specialists. The speed specialist also results in a problematic interaction. A pre-condition of movement is that power be available to the coil-car. An effect of the speed-specialist is that the power consumption of the coil-car will increase and possibly lead to the lack of power, thereby blocking the movement pre-condition. Again this is spotted during retrieval and the blocked-precondition is also marked as applicable. In conclusion, the base case is judged to be adaptable. In a real retrieval session its precise adaptability would be computed in terms of the number of specialists and strategies needed and this measure would be used to discriminate among alternative adaptable cases.

Apart from the benefits of retrieving adaptable cases this method also offers more that just a case and a similarity measure to adaptation. It also offers a representation of the nature of the similarities and dissimilarities between the target and base in the form of the specialists and strategies that are deemed applicable. This additional knowledge is very useful during adaptation in pointing out precisely *what* needs to be adapted and *how* it may be adapted.

4 Experiments

The following three sets of experimental data demonstrate the accuracy and performance characteristics of adaptation-guided retrieval. All of them were run using the Déjà Vu system. Experiment 1 tests the retrieval accuracy of AGR versus a standard similarity model of retrieval. Experiment 2 examines the relationship between retrieval cost and the size of the case-base in AGR. Finally, Experiment 3 looks at the overall performance of AGR compared to a standard similarity model.

4.1 Experiment 1: Accuracy of Retrieval

Traditional approaches to retrieval select cases on the basis of semantic similarity in the hope that they will also be the most adaptable. Experiment 1 shows that this assumption can be unwarranted. The standard similarity model (SS) used was a classical, distance-based similarity metric that compares features on the basis of their separation distance in the knowledge-base; for example, identical matches obtain perfect similarity, objects that shared a common parent obtain less similarity, and objects that share a common grandparent fair worse again.

Trials were carried out with two different case-bases (see Figure 5). Trial 1 used a case-base that contained 45 cases all from the same plant-model; that is, the same track layout and plant objects were used in each case. The same plant model was also used for the 45 target problems of the first trial (see e.g., Figure 1). The most adaptable case in the case-base was computed for each target problem. The accuracy of the two retrieval methods was then measured for the 45 targets. The results showed that AGR selected the most adaptable case 100% of the time whereas the SS method was only accurate 70% of the time; this difference was statistically significant (see Trial 1 in Figure 5 ;$chi^2(1) = 14.295$, $p < .0001$).

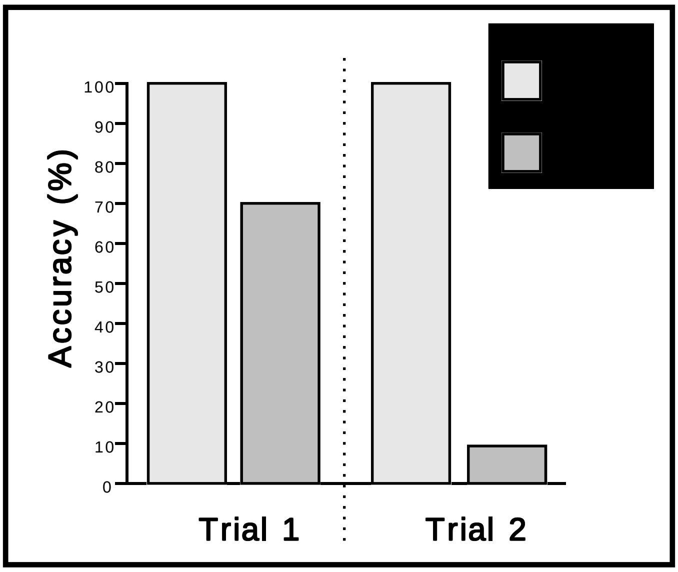

Figure 5. Expt. 1: The Relative Accuracy of Two Retrieval Methods

In the second trial, we used a case-base containing 120 cases involving 8 different plant-models. Various plant models were also used in the 45 target problems employed. The SS method fares even worse on this more realistic case-base; AGR was still 100% accurate, but SS decreased to 12% (see Trial 2 in Figure 5; $chi^2(1) = 65.34$, $p = .0001$). The standard similarity method degrades because it selects cases from the same plant-model rather cross-model cases. It is mislead by exact entity matches between the target and cases involving the target's plant-model even though cases from different plant model are often easier to adapt.

Clearly, the standard similarity method could be improved with a more elaborate weighting scheme to closer approximate the concept of adaptability. However, such improvements would implicitly include the knowledge that is explicitly used in the AGR method. Furthermore, the tailoring of similarity metrics is a complex trial and error process that depends greatly on the current state of the system. Finally, as we shall see in the other experiments, it is not clear that such remedial adjustments actually result in any computational gain over AGR (see Experiment 3).

4.2 Experiment 2: Avoiding Swamping Problems

The AGR method is clearly more complicated than standard similarity methods. It is, therefore, important to ascertain whether it is particularly prone to a special case of the utility problem in CBR systems, known as the *swamping problem* (see Francis & Ram, 1993). Utility problems occur when the uncontrolled accumulation of knowledge results in a performance degradation because the cost of locating relevant knowledge is (on average) more than the saving obtained in using this knowledge. In many CBR systems the swamping problem arises because the cost of retrieval is directly proportional to the number of cases in the case-base; as a case-base expands overall problem solving performance may actually degrade.

One solution to this problem is to limit retrieval time; the best case found within a given time limit is selected (Veloso, 1992). This solution invariably results in the retrieval of a sub-optimal case and, of course, such sub-optimal cases may be difficult or impossible to adapt. Adaptation-guided retrieval is less prone to the swamping problem, because the cost of retrieval does not depend on the size of the case-base as a whole but more on the number of cases that are adaptable (relevant) to the target problem; the base-filtering stage of retrieval ensures that non-adaptable cases are not examined during retrieval. The avoidance of swamping in AGR is illustrated in Experiment 2.

In Experiment 2, we varied the size of the case-base from 30 to 120 cases in units of 30. Twenty target problems were tested on each of these case-bases. Figure 6 shows the mean retrieval times for the test problems in three different conditions. The *standard condition* shows the performance of the system on the test problems. Note that while there is an increase in retrieval time, it is not linear with respect to the total size of the case-base. Rather it is linear relative to the number of adaptable cases found (in this experiment roughly 10% or less of the total case-base). In Figure 6 the numbers beside the boxes of the standard curve indicate the number of adaptable cases on each retrieval. The *constant condition* proves this point, by holding the number of adaptable cases in the case-base constant for each target (3 adaptable cases were used in each case-base). When the number of adaptable cases is fixed, the curve flattens relative to the standard condition.

Of course, it could be argued that the linear increase in the standard condition is still unacceptable. So, in the *bounded condition*, we examined performance by terminating retrieval when the first adaptable case is found (rather than the most adaptable case). This bounded retrieval method works well in that retrieval time remains flat irrespective of the overall case-base size or the number of adaptable cases available. We should, however, remember that this version of the system does not retrieve the most adaptable case, so there may be more processing overhead in the adaptation stage.

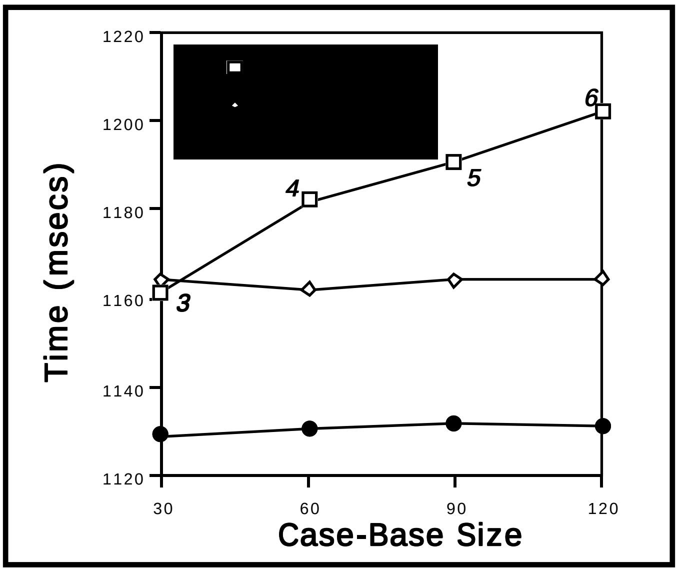

Figure 6. Swamping Experiments

These results show that the swamping problem is not a major issue for AGR. AGR's performance advantage is due to the fact that it only considers adaptable cases during retrieval and that these cases can be very quickly located by the indexing scheme offered by the adaptation knowledge. Many CBR approaches employ base-filtering methods to cut down the number of cases considered during retrieval but many of these approaches are either over general, and still select many more than just the relevant cases, or they are over specific and tend to ignore some easily adapted cases.

4.3 Experiment 3: Overall System Performance

In Experiments 1 and 2 we just considered retrieval. However, AGR should also have benefits for overall system performance (i.e., combining retrieval and adaptation). We have seen that AGR's retrieval accuracy is very respectable relative to a standard similarity model. AGR should be more accurate and faster in the adaptation stage because the retrieval stage identifies what adaptation knowledge should be used. In Experiment 3, we examined the effect of adaptation-guided retrieval on the overall problem-solving time. Two versions of Déjà Vu were used; one that used AGR (the AGR-system) and another that used semantic similarity-based retrieval and an adaptation component (the similarity-based, adaptation system or SBA-system). Each system had the same case-base of 100 cases, the same adaptation knowledge and was tested with the same 45 target problems.

Figure 7 shows the cumulative solution times for the two systems over the 45 problems (problems were roughly ordered in terms of their complexity). The AGR-system was considerably better than the SBA-system taking only 120 seconds to solve the 45 problems compared to 280 seconds in the SBA-system. The mean solution time for problems in the AGR-system (*M* = 2.07 secs; *SD* = 2.07) is about three times faster that in the SBA-system (*M* = 6.22 secs; *SD* = 5.66); a difference that is statistically reliable ($t(44) = 5.65$, $p < .0001$).

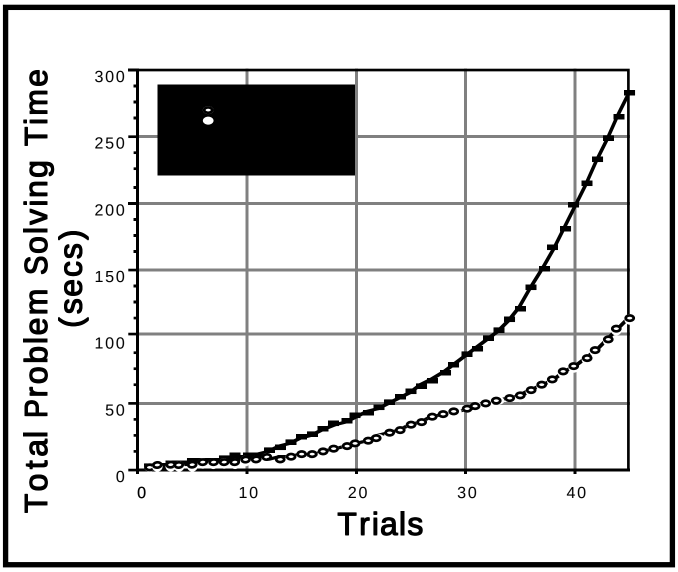

Figure 7. Expt. 3: Solution Times for Two Systems

The performance of the AGR-system is much better than the SBA-system because it retrieves the most adaptable case and locates the relevant adaptation knowledge for this case during retrieval. Furthermore, the benefits of AGR emerge most strongly when problems become more complex, because the sketchy nature of standard, similarity-based retrieval has a greater tendency to be mislead.

5 Conclusions

Many researchers have been attracted the idea of adaptation-guided retrieval but have worried about its computational efficiency. In this paper, we have tried to show that these worries are unfounded. First, the explicit use of adaptation knowledge ensures that the most adaptable case is always retrieved with a greater retrieval accuracy than more conventional approaches (see Expt. 1). Second, adaptation-guided retrieval maintains the cost of retrieval at an acceptable level or can be bounded to achieve near-constant retrieval times (see Expt. 2). So, the technique should scale well on larger case-bases. Third, the overall adaptation costs are greatly reduced because the most adaptable case is always selected and preliminary adaptation work is performed during retrieval. So, there are considerable performance improvements in the overall cost of problem solving (see Expt. 3).

In addition, the closer integration of retrieval and adaptation provides a much more flexible CBR model. With conventional approaches, changes to the adaptation capabilities of a system are not immediately reflected in the retrieval preferences of the system. Instead changes must be made to the retrieval heuristics in order to capture the new adaptation possibilities. In contrast, because the retrieval and adaptation stages are directly coupled in Déjà Vu, any changes to its adaptation capabilities *will* be immediately available to the retrieval system; this is because the altered adaptation knowledge itself is used explicitly in retrieval.

We acknowledge that standard similarity models of retrieval and standard CBR architectures could be parametrically varied to improve the performances we have shown here. However, we doubt whether any such systems could better the results found for the AGR-system. Furthermore, the likelihood is that any such system would be merely trying to mimic AGR within the confines of standard approaches.

Finally, the representational requirements of the approach are domain independent and thus facilitate the adoption of the technique across a range of CBR application domains. Already Déjà Vu has been used to investigate a number of different software design domains. As well as plant-control software, a Motif graphical user interface design has also been investigated. Initial results suggest that AGR transfers well to this quite different software design domain.

6 References

Bareiss, R., and King, J. A. (1989). Similarity Assessment in Case-based Reasoning. Proceedings of the Case-Based Reasoning Workshop, (pp. 67 - 71). Florida, U.S.A.

Birnbaum, L., Collins, G., Brand, M., Freed, M., Krulwich, B., Pryor, L. (1989) A Model-Based Approach to the Construction of Adaptive Case-Based Planning Systems. Proceedings of the Case-Based Reasoning Workshop, (pp. 191 - 198). Florida, USA.

Cain, T., Pazzani, M. J., and Silverstein, G. (1991) Using Domain Knowledge to Influence Similarity Judgements. Proceedings of the Case-Based Reasoning Workshop, (pp. 191-198). Washington D.C., U.S.A.

Francis, A.G., & Ram, A. (1993) Computational Models of the Utility Problem and their Application to a Utility Analysis of Case-based Reasoning. Proceedings of the Workshop on Knowledge Compilation and Speed-Up Learning.

Goel, A. (1989) Integration of Case-Based Reasoning and Model-Based Reasoning for Adaptive Design Problem Solving. Ph.D. Thesis. Ohio State University, Usa.

Hammond, K. J. (1989). Planning from Memory. New York: Academic Press.

Hendler, J., Tate, A., Drummond, M. (1990) AI Planning: Systems and Techniques. AI Magazine, **11**(2), (pp. 61-77).

Kolodner, J. (1989). Judging Which is the "Best" Case for a Case-Based Reasoner. Proceedings of the Case-Based Reasoning Workshop, (pp. 77 - 84). Florida, U.S.A.

Rissland, E. L., and Ashley K. D. (1988). Credit Assignment and the Problem of Competing factors in Case-Based Reasoning. Proceedings of the Case-Based Reasoning Workshop. (pp. 327 - 344) Florida, U.S.A.

Smyth, B., & Keane, M. (1994). Retrieving Adaptable Cases: The Role of Adaptation Knowledge in case Retrieval. Topics in Case-Based Reasoning, Springer Verlag (pp. 209-220).

Smyth, B., & Cunningham, P. (1992a). Déjà Vu: A Hierarchical Case-Based Reasoning System for Software Design. Proceedings of the 10th European Conference on Artificial Intelligence. (pp. 587 - 589). Vienna, Austria

Smyth, B., & Cunningham, P. (1992b). Déjà Vu: A Recursive Case-Based Reasoning System for Software Design. Proceedings of the 5th Irish Conference on Artificial Intelligence and Cognitive Science (pp. 587 - 589). Limerick, Ireland.

Sycara, K. P., & Navinchandra, D. (1991). Influences: A Thematic Abstraction for Creative Use of Multiple Cases. Proceedings of the Case-Based Reasoning Workshop. (pp. 133 - 144) Washington, D.C., U.S.A.

Thagard, P, Holyoak, K.J., Nelson, G. & Gochfeld, D. (1990). Analog Retrieval by Constraint Satisfaction. Artificial Intelligence, 46, (pp. 259-310).

Veloso, M. (1992) Learning by Analogical Reasoning in General Problem Solving. Ph.D Thesis (CMU-CS-92-174). Carnegie Mellon University, Pittsburgh, USA.