Learning to Refine Indexing by Introspective Reasoning*

Susan Fox and David B. Leake

Computer Science Department
Indiana University
Bloomington, IN 47405, USA

Abstract. A significant problem for case-based reasoning (CBR) systems is determining the features to use in judging case similarity for retrieval. We describe research that addresses the feature selection problem by using introspective reasoning to learn new features for indexing. Our method augments the CBR system with an introspective reasoning component which monitors system performance to detect poor retrievals, identifies features which would lead retrieval of more adaptable cases, and refines the indexing criteria to include the needed features to avoid future failures. We explore the benefit of introspective reasoning by performing empirical tests on the implemented system. These tests examine the effect of introspective index refinement, and the effects of problem order on case and index learning, and show that introspective learning of new index features improves performance across the different problem orders.

1 Introduction

Selecting the best set of features to use in indexing a case-based reasoning (CBR) system's memory may be difficult; determining which features are most appropriate may require experience [15]. An appealing alternative is to permit the system itself to learn what features are relevant in response to its experiences. Our research uses introspective reasoning to permit a CBR system to detect indexing problems and to alter its indexing criteria to correct the problems by learning to consider new features. Our method applies introspective reasoning to learn features implicit in the original problem description but not used explicitly in the CBR system's initial indexing scheme, in order to direct retrieval towards cases that can easily be adapted.

Our testbed computer system, ROBBIE[2], is a case-based route planner. Its CBR component learns by adding successful plans to its memory, as in other case-based route planners such as ROUTER [13]. However, our research focus is not the route planning task itself. Instead, the focus is on how introspective reasoning can be used by the CBR system to refine its indexing criteria. ROBBIE combines a case-based planner with an introspective component. The introspective component contains a declarative model of

* This work was supported in part by the National Science Foundation under Grant No. IRI-9409348. To appear in *Proceedings of the First International Conference on Case-Based Reasoning*, Sesimbra, Portugal, 1995.

[2] Re-Organization of Behavior By Introspective Evaluation

the *ideal* reasoning behavior of the system. The model provides expectations about the ideal case-based planning performance; these expectations are compared to the actual reasoning performance to discover discrepancies. A discrepancy indicates a reasoning failure; the failure is corrected by explaining why it occurred and revising the reasoning process to avoid future failures. Using introspective learning to improve the case-based reasoning process itself was first suggested by Birnbaum et al. [6].

Introspective improvement of reasoning processes is a relatively new approach, and there have been few evaluations of its effect on the performance of a system using it. We designed empirical tests to examine the effect of introspective learning on ROBBIE's performance, and to examine the effect of problem order on both case learning and index learning. Our results show that introspective reasoning for learning indexing criteria improves performance *and* efficiency across different problem orders.

We will first present the ROBBIE system and discuss its mechanisms for case-based and introspective learning, providing an example of its learning to illustrate the process. We will then describe our experiments and what we can conclude from them.

2 The ROBBIE System

The ROBBIE system demonstrates the application of an introspective self-model to the task of refining the indices used to retrieve cases from memory. The system has two main parts: a planner which develops plans through case-based reasoning [2, 14, 15] and applies them through execution in a simulated world, and an introspective model-based reasoner which detects, explains, and repairs reasoning failures caused by poor indexing criteria.

The performance task of ROBBIE is to navigate around a simulated set of city streets as a pedestrian. ROBBIE is situated in a "map" of streets and learns new routes to navigate around that map. ROBBIE is given a goal location and must create a plan for getting from its current location to the goal. Plan creation involves retrieving and adapting an old case. Because adaptation is not always guaranteed to be correct, ROBBIE executes the newly created plan using reactive planning [8] to fill in missing details and recover from failures. Execution evaluates the quality of the CBR-created plan, and permits ROBBIE to arrive at a solution despite flaws in the plan. This provides a form of execution-time case adaptation: the reactive planner can alter the plan given it by removing useless steps and adding new steps as necessary. In addition, the reactive planner handles facets of the simulated world that cannot be predicted by the CBR system, such as traffic lights and blocked streets. The plans resulting from execution are stored in memory for future use. The plans in memory are indexed by their starting and ending locations, plus features discovered through introspective learning. Plan retrieval in ROBBIE is a two-stage process: first ROBBIE gathers a pool of generally similar candidate cases, then it selects the best candidate from that pool.

ROBBIE's high-level task is to improve its reasoning process (in particular, its feature selection process) when it detects failures in its reasoning. As the planner reasons about *its* task, the introspective component monitors its reasoning process by comparing it to a declarative model describing the planner's ideal reasoning processes [6, 9, 12]. Reasoning failures occur when the model's expectations about the reasoning process are

not fulfilled by the actual reasoning. Expectations are encoded as "assertions;" facts that would be ideally true of particular points in the case-based reasoning process. For example, one assertion states that *the index for retrieval will be most similar to the index of the best case in memory for adapting to the current problem*. This assertion reflects the desire to retrieve easily adaptable cases.

Some assertions describe the ideal behavior of specific implementation details for ROBBIE. Others describe ideal aspects at an abstract level, in terms of control and information flow through the CBR process. The assertion in the previous paragraph is an example of an abstract assertion. A specific one would describe, for instance, how similarity judgements are made and the ideal result. The assertions are clustered together by the part of the reasoning process to which they refer, and by how abstract or specific they are. Each assertion also contains links to other causally related assertions inside *or outside* its cluster. Fox and Leake [11] contains a detailed description of the model and its representation.

Failure detection for the introspective reasoner involves noticing discrepancies between assertions and actual behavior. At each point in the reasoning process, the introspective reasoner compares the actual reasoning to the ideal. When they fail to match, a reasoning failure has occurred (See [9] for a more extensive discussion of ROBBIE's failure detection). Detecting failures due to faulty indexing criteria requires monitoring the entire reasoning process: the failure may not be discovered until after the inappropriately retrieved case has been adapted and executed. The problem is compounded by the desire to recognize not only poor retrievals which lead to a failed solution, but poor retrievals that, despite leading to a successful solution, waste effort in adaptation or execution.

ROBBIE's introspective reasoner uses hindsight to detect when a successful solution was based on the wrong case (indicating wasted effort in adaptation and execution). After a successful plan has been generated, it is used as a standard for determining which case *should* have been retrieved. The final solution is compared to the plans in memory to determine whether the plan that was retrieved was really the most similar plan in memory. The goal to retrieve the most similar plan is captured in the model by the expectation that the final solution should be most similar to the *solution* of the retrieved case.

If the wrong plan has been retrieved, causing the adapted plan to contain wasted steps, execution of a plan will eliminate wasted steps, leading to a streamlined final solution. In storing a new solution, ROBBIE looks for the case in memory with the closest solution, according to the similarity of their plan steps (which may be quite different than when retrieved). If the final solution is now similar to another case, then a retrieval failure is indicated.

Once a reasoning failure is detected the model is used to create an explanation of the failure in terms of other failed assertions, and to suggest a repair. An explanation is formed by searching the model for other causally-related assertion failures. Other assertion failures might have been previously undetected, but may be suspected in the context of the detected failure. The introspective reasoner is guided in its search for an explanation by the links between assertions that are causally related.

Attributing a detected failure to faulty indexing criteria depends on the discovery

of some relevant feature of the new solution and its closest matching case in memory whose explicit inclusion in the indexing criteria would have permitted a correct retrieval (i.e., retrieval of the case closer to ROBBIE's final solution). An assertion about the index formation process, when re-examined in light of a failure, is evaluated by looking for missing features which fit the above requirement. In searching for shared features, ROBBIE currently uses a set of abstract feature types, such as "starts-at X" and "moves-in-direction Y," which it attempts to instantiate for the current cases. ROBBIE looks for instantiations of feature types that apply to both the solution and the best-match case that was not retrieved, and that are not currently used in indexing. The feature types are ranked by a pre-defined importance hierarchy; if more than one instantiated type applies to the two cases, the more important one is selected to add to the indexing criteria.

Figure 1 illustrates an implemented example of ROBBIE's introspective index refinement. In this situation, ROBBIE has in memory plan A, which describes how to get from L1 to L2, and plan B, which describes how to get from L2 to L3. The current task is to get from L4 to L2. ROBBIE's initial indexing uses the geographic closeness of starting and ending locations; by this measure, plan A appears to be the closest because it shares the same ending location (ROBBIE's retrieval criteria do not include reversals of known routes, so plan B does not look similar at all). Plan A is selected and adapted to create plan C (dashed line in Fig. 1). During this process, the introspective reasoner compares the system's reasoning process to its ideal model, but cannot detect a problem.

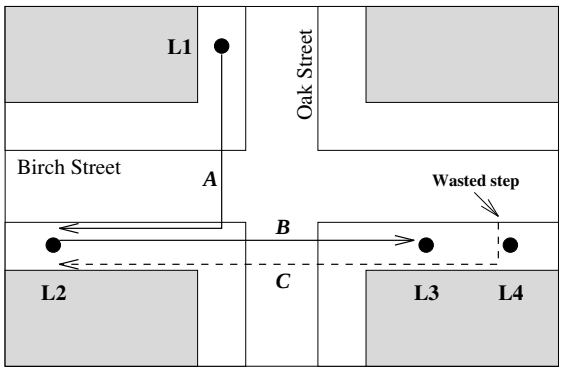

Fig. 1. example of ROBBIE's introspective learning

However, adaptation of plan C during execution eliminates the wasted step; moving south across the sidewalk makes no progress towards the goal. The final solution is a plan which moves straight west. When the resulting plan is stored in memory, the introspective reasoner detects a reasoning failure: the final solution is more similar to plan B than to plan A, the retrieved plan.

In explaining this failure, the introspective reasoner must search for other causally-related assertion failures. Some such assertions relate to the indexing criteria used. One assertion related to the original failure is that *the index for retrieval will include all the features needed to retrieve the closest case*. This assertion is assumed true unless another

failure has been detected: if nothing appears wrong, we may assume that the current indices are sufficient. In the context of an existing failure, this assertion is examined by considering the features of the final solution and best matching case to determine if a feature exists which would enable avoiding the faulty retrieval. In this case a feature is found: that B and C both involve moving straight along an east/west street. This assertion suggests a repair to the indexing criteria: adding the feature "moves straight east/west" to the features used in indexing cases, and re-indexing the cases in memory to include the new feature.

3 Empirical Evaluation

There has been growing interest in evaluation of case-based reasoning systems to identify to what extent they succeed and what components of their processing are responsible for their success [1]. However, little work has been done to quantify the benefits of introspective reasoning for improving reasoning methods, and the relationship between case presentation order and learning is largely unaddressed.

We performed experiments to evaluate the effect of introspective learning of new index features on the overall performance of ROBBIE. We chose a range of situations in which to test ROBBIE's performance which would also demonstrate the effect of the problem order on the system's learning. Such tests should show that introspective re-indexing improves performance across the range of situations, that problem order had a distinct and dramatic effect on overall performance, and that introspective re-indexing causes improved performance even as quality of problem order declines.

It is acknowledged that problem order can strongly affect learning in general, and learning of CBR systems in particular (e.g. [4, 18]). A sequence of goals which gradually increases in complexity and distance from the original cases in memory, and which covers the range of possible situations, should maximize the effectiveness of the case-based reasoner's learning.

Guided by the principles outlined in the previous paragraph, we created by hand a "well-ordered" sequence which included 40 randomly generated goal locations on a set of streets, each appearing exactly once. In these tests, ROBBIE began at the first location, then moved from one to the next in sequence. From this ordered sequence of locations we derived five groups containing five sequences in each. For four of the groups, we permuted the original sequence by randomly swapping a percentage of its locations with other locations in the sequence. The groups had 25%, 50%, 75%, and 100% of their elements swapped. The final group contained random arrangements of the 40 goal locations. We did not attempt to create a deliberately bad sequence, although that would be an interesting extension of this experiment.

To examine the effect of introspective re-indexing, we compared the performance of ROBBIE on runs using just case learning with runs using case and introspective learning. We measured the number of successfully completed plans at the end of a test run and compared the results across runs and sequences. We also measured the percentage of cases in memory considered for each retrieval. Learning new index features increases the focus of retrieval; the decrease in percentage of cases considered over time forms a measure of retrieval efficiency. Because ROBBIE chooses randomly between two alter-

natives that are considered equally good, we ran each sequence twenty times under the same initial conditions to factor out the effect of individual random decisions.[3]

The initial case memory for these tests was very small to examine the effects of learning under difficult, knowledge-poor conditions. The initial cases in memory were clustered in one section of the set of streets, to examine how a learner with limited initial information could expand its knowledge to encompass its domain. The model used during testing contained a restricted set of assertions to streamline the diagnosis process.

We predict the rate of success should be highest for the initial "well-ordered" sequence, and the rate (with and without introspective re-indexing) should decline as sequences become more random. For each sequence, runs with introspective re-indexing should be more successful than those without. We do not expect introspective re-indexing to overcome the negative effect of poor problem order Learning new indexing features depends on the right features being available in cases in memory. The most difficult scenario for introspective re-indexing occurs when the initial memory size is small and the problem order produces poor performance: the smaller the case memory, the fewer the opportunities to learn useful new features for indexing.

We expect introspective re-indexing to improve retrieval efficiency by causing fewer cases to be considered in retrieval, as newly learned features allow finer-grained distinctions between cases. However, there is a possible danger in this restriction: when the system fails repeatedly and memory remains small over time, re-indexing could cause over-restriction of retrieval, leading to failures when applicable cases are ignored. Consequently, another question we studied was whether index learning decreases the planner's success rate by over-restricting the retrieval process.

4 Results

Success rate was measured by counting the number of successful plans created for each run of a sequence. Figure 2 illustrates the overall success rate of each group of sequences used. The success rate was averaged across each of the 20 runs of a sequence, then the results for each sequence in a group were averaged again. The trend as sequences became less ordered is as expected: performance drops as problems become less easy to match against known cases. The combination of case learning and introspective re-indexing of memory led to better performance than case learning alone for every group.

Figure 3 shows the success rate for each sequence individually. The tendency for performance to degrade with less ordered sequences can still be seen, and for all but one sequence the addition of introspective learning proved beneficial (we discuss the anomalous case below). The wide range of behaviors within each group warrants further explanation in the future: we describe a possible correlation with retrieval efficiency in the next section.

We measured the change in retrieval efficiency by considering the percentage of stored plans ROBBIE considered similar for each retrieval during a test run. We graphed the course of the run on the x-axis by counting each retrieval in sequence as a point in time.

[3] ROBBIE was presented with a total of 41,600 goals: 26 sequences, each 40 goals long, and 20 runs with introspection, 20 without.

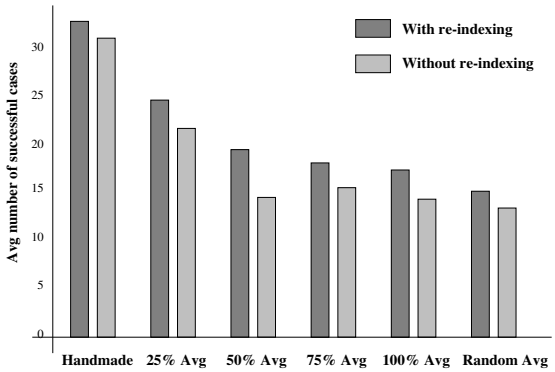

Fig. 2. Average success rate declines as order becomes more random; introspective reasoning slows the decline

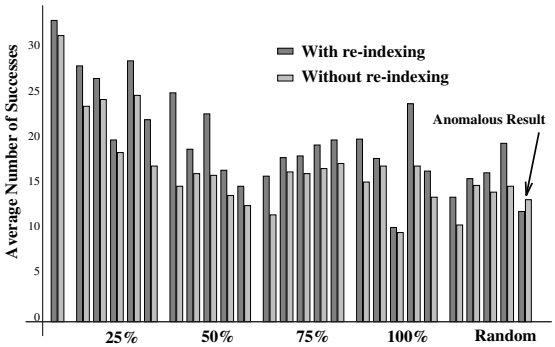

Fig. 3. Average success rate for each sequence; one anomaly demonstrates over-constraint of retrieval

We considered "successive averages" of the percentage considered on the y-axis: each value was the average of the percentages from the beginning of the sequence to that point. By putting runs of a sequence together on one set of axes, we see trends in the retrieval performance. For a more detailed discussion of these results see [10].

Figure 4 shows typical results for a sequence: the averages considered with re-indexing decrease over time to a much lower level than those without: 30-45% instead of 60-70%. A similar pattern appeared for every sequence except the anomalous one. In some cases the decrease with re-indexing was much more dramatic. The size and consistency of the drop were greater for sequences with high success rates, and for sequences with large differences between success rates with and without re-indexing.

The anomalous sequence led to poor performance under any measure. There was no apparent difference between the retrieval efficiency with re-indexing or without it, suggesting that the new features learned introspectively did not apply to later situations and so could not provide any benefit. The new features did not group together useful cases, but could occasionally cause an additional failure by restricting retrieval so that a useful case might not be considered at all. This limiting effect on re-indexing would be mag-

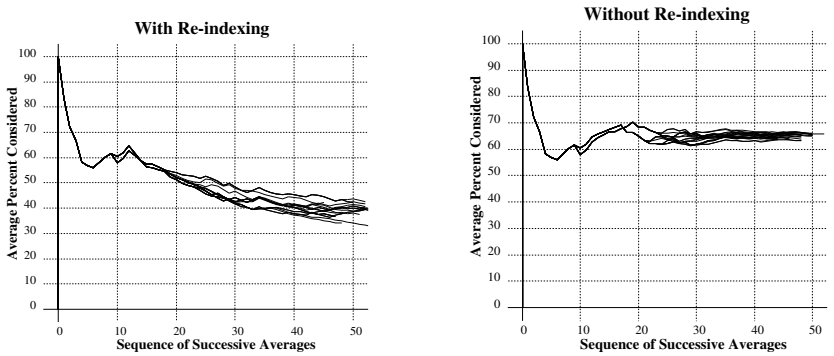

Fig. 4. Successive average of retrieval: with re-indexing, efficiency improves by considering fewer cases on average (50% permuted sequence)

nified by the small size of the case base, which in this case remained small throughout the run as few goals were successfully achieved. Characterizing sequences like this one further and determining their frequency is an important task for the future.

5 Related Work

Other work has combined CBR with learning about the reasoning process itself. A proposal by Birnbaum et al. [6] inspired ROBBIE's framework, but that proposal was not implemented and did not use a hierarchical model. Meta-AQUA [7] performs failure-driven introspective learning, but contains no explicit model of correct behavior. Instead, it uses a set of template descriptions for reasoning failures which provide diagnosis and repair information. Autognostic [20] applies an existing framework for device modeling (SBF) to introspective modeling of the route planner ROUTER; their approach focuses on explanation of failures without strongly addressing failure detection. IULIAN [17] also performs introspective learning, but instead of using an explicit introspective model, it stores plans for repairing its reasoning which are recalled when failures occur. Arcos and Plaza [3] create a unified architecture for describing case-based and meta-level problem solving tasks by describing each process by decomposition into tasks and sub-tasks. IDEAL [5] investigates index learning, but uses a model of its domain, not the reasoning task, as well as cases from its domain, in order to learn new indices.

The focus of our research, by comparison, is the development of a framework for modeling reasoning processes which can address both detection and repair of reasoning failures. We believe the framework, using a model of ideal reasoning behavior to detect and explain failures, has wide applicability, but so far it has been applied only to the task of learning indexing criteria for retrieving plans in ROBBIE.

6 Conclusions

Using a model of reasoning behavior to detect and repair reasoning flaws permits a system to learn from experience how to reason best in its environment. We have applied this idea to the learning of indexing criteria for case retrieval. We propose a framework which combines case-based learning with an introspective reasoning component. The introspective reasoner uses a model of expected ideal reasoning performance to detect discrepancies in actual performance, and to diagnose and repair such expectation failures. ROBBIE implements this framework for a case-based planner that learns new plans as it creates them and introspectively learns which features are required for retrieving easily adaptable cases from memory. Because the quality of a retrieval scheme depends on retrieving cases that are easy to adapt, it is desirable to achieve more direct connections between similarity criteria and case adaptation ability in CBR systems (e.g., [16, 19]); our research is one way to make indexing criteria better reflect adaptability. In addition, introspective reasoning for self-improvement is a general approach that could be extended to other portions of the CBR process: altering adaptation, evaluation, and case storage (See [9] for a more general discussion).

Little empirical evaluation has been done to measure the effects of introspective reasoning on the behavior of systems which use it. We designed a set of empirical tests of ROBBIE to examine the effect of introspective learning of new index features by comparing its performance with and without index refinement. In addition, we examined the effect of problem order on the performance of case and index learning. It is well known that the order in which problems are presented to a learning system can have a large effect on the quality of the learning that the system performs. Case-based learning systems appear particularly susceptible to this effect.

We found that, even under knowledge-poor initial conditions, introspective learning of new index features improved the success rate of the system over case learning alone, while at the same time improving the efficiency of the retrieval process by focusing retrieval on the best cases. Problem order had a large effect on the learning of the system, but regardless of problem order the inclusion of introspective re-indexing outperform case learning alone. One anomalous case demonstrated that under difficult conditions re-indexing may cause over-constraint of the retrieval process, causing it to ignore applicable cases. That over-constraint, made worse by poor case learning performance, led to better performance without re-indexing than with it.

By gradually increasing the complexity of situations a case-based system faces, we can guide its learning to be more productive. The addition of introspective learning of index features also improves performance and eases the burden on the system designer to pick the right initial features. Further exploration of what "well-ordered" means for case-based systems is required, but extending ROBBIE to introspectively alter other portions of the CBR process is the next step.

References

1. D. Aha, editor. *Proceedings of the AAAI-94 Workshop on Case-Based Reasoning*, Seattle, WA, July 1994.
2. R. Alterman. An adaptive planner. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 65–69, Philadelphia, PA, August 1986. AAAI.

3. J. Arcos and E. Plaza. A reflective architecture for integrated memory-based learning and reasoning. In S. Wess, K.D. Altoff, and M. Richter, editors, *Topics in Case-Based Reasoning*. Springer-Verlag, Kaiserslautern, Germany, 1993.
4. W.M. Bain. *Case-based Reasoning: A Computer Model of Subjective Assessment*. PhD thesis, Yale University, 1986. Computer Science Department Technical Report 470.
5. S. Bhatta and A. Goel. Model-based learning of structural indices to design cases. In *Proceedings of the IJCAI-93 Workshop on Reuse of Design*, Chambery, France, September 1993. IJCAI.
6. L. Birnbaum, G. Collins, M. Brand, M. Freed, B. Krulwich, and L. Pryor. A model-based approach to the construction of adaptive case-based planning systems. In R. Bareiss, editor, *Proceedings of the Case-Based Reasoning Workshop*, pages 215–224, San Mateo, 1991. DARPA, Morgan Kaufmann, Inc.
7. M. Cox and A. Ram. Managing learning goals in strategy-selection problems. In *Proceedings of the Second European Workshop on Case-Based Reasoning*, pages 85–93, Chantilly, France, 1994.
8. R. J. Firby. *Adaptive Execution in Complex Dynamic Worlds*. PhD thesis, Yale University, Computer Science Department, 1989. Technical Report 672.
9. S. Fox and D. Leake. Using introspective reasoning to guide index refinement in case-based reasoning. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, pages 324–329, Atlanta, GA, 1994. Lawrence Erlbaum Associates.
10. S. Fox and D. Leake. An introspective reasoning method for index refinement. In *Proceedings of 14th international Joint Conference on Artificial Intelligence*. IJCAI, 1995.
11. S. Fox and D. Leake. Modeling case-based planning for repairing reasoning failures. In *Proceedings of the 1995 AAAI Spring Symposium on Representing Mental States and Mechanisms*, Stanford, CA, March 1995. AAAI. (ftp.cs.indiana.edu:/pub/leake/p-95-02.ps.Z).
12. M. Freed and G. Collins. Adapting routines to improve task coordination. In *Proceedings of the 1994 Conference on AI Planning Systems*, pages 255–259, 1994.
13. A. Goel, K. Ali, and Andrés Gómez de Silva Garza. Computational tradeoffs in experience-based reasoning. In *Proceedings of the AAAI-94 workshop on Case-Based Reasoning*, pages 55–61, Seattle, WA, 1994.
14. C. Hammond. *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press, San Diego, 1989.
15. J. Kolodner. *Case-Based Reasoning*. Morgan Kaufman, San Mateo, CA, 1993.
16. D. Leake. Constructive similarity assessment: Using stored cases to define new situations. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pages 313–318, Bloomington, IN, 1992. Cognitive Science Society.
17. R. Oehlmann, P. Edwards, and D. Sleeman. Changing the viewpoint: Re-indexing by introspective questioning. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, pages 675–680. Lawrence Erlbaum Associates, 1994.
18. M. Redmond. *Learning by Observing and Understanding Expert Problem Solving*. PhD thesis, College of Computing, Georgia Institute of Technology, 1992. Technical report GIT-CC-92/43.
19. B. Smyth and M. Keane. Retrieving adaptable cases: The role of adaptation knowledge in case retrieval. In S. Wess, K. Althoff, and M Richter, editors, *Topics in Case-Based Reasoning*, pages 209–220, Berlin, 1994. Springer Verlag.
20. E. Stroulia and A. Goel. Task structures: What to learn? In M. desJardins and A. Ram, editors, *Proceedings of the 1994 AAAI Spring Symposium on Goal-driven Learning*, pages 112–121. AAAI Press, 1994.

This article was processed using the LaTeX macro package with LLNCS style