To appear in *Proceedings of the Second European Workshop on Case-Based Reasoning*. Chantilly, France. Nov 7-10, 1994.

Managing Learning Goals in Strategy-Selection Problems

Michael T. Cox and Ashwin Ram
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
{cox,ashwin}@cc.gatech.edu

Abstract

In case-based reasoning systems, several learning techniques may apply to a given situation. In a failure-driven learning environment, the problems of *strategy selection* are to choose the best set of learning algorithms or strategies that recover from a processing failure and to use the strategies to modify the system's background knowledge so that the failure will not repeat in similar future situations. A solution to this problem is to treat learning-strategy selection as a planning problem with its own set of goals. Learning goals, as opposed to ordinary goals, specify desired states in the background knowledge of the learner, rather than desired states in the external environment of the planner. But as with traditional goal-based planners, management and pursuit of these learning goals becomes a central issue in learning. Examples are presented from a multistrategy learning system called Meta-AQUA that combines a case-based approach to learning with nonlinear planning in the knowledge space.

1 Introduction

As case-based reasoning (CBR) research addresses more sophisticated task domains, the associated learning issues involved become increasingly complex. Multistrategy learning systems attempt to address the complexity of such task domains by bringing to bear many of the learning algorithms developed in the last twenty years. Yet the goal of integrating various learning strategies is a daunting one, since it is an open question as how best to combine often conflicting learning-mechanisms. This paper examines the metaphor of goal-driven problem-solving as a tool for performing this integration. Learning is viewed as simply another problem to solve; the learning problem is formulated by posting learning goals (such as goals to answer a question or to reconcile two divergent assertions). A plan is assembled by choosing various learning algorithms from the system's repertoire that can achieve such goals.

In formulations with conjunctive goals, however, numerous difficulties arise such as goal conflicts, protection intervals, and many-to-many relationships between goals and algorithms. For instance, a familiar goal conflict in planning systems is the "brother-clobbers-brother" goal interaction (Sussman, 1975), whereby the result of one plan that achieves a particular goal undoes the result or precondition of another plan serving a different goal. If a learning goal specifies a state change to the background knowledge of the system, rather than a state change in the world, then learning plans can have similar effects. Changes to specific knowledge may affect previous changes to the background knowledge performed by other learning algorithms.

This paper illustrates such interactions with a multistrategy learning system called Meta-AQUA. The performance task in Meta-AQUA is story understanding. That is, given a stream of concepts as the representation for a story sequence, the task is to create a causally connected conceptual interpretation of the story. The story-understanding strategies available to the system are CBR, analogy, and explanation. If the system fails at its performance task, its subsequent learning-subtasks are (1) *blame assignment* — use case-based methodologies to analyze and explain the cause of its misunderstanding by retrieving past cases of meta-reasoning, (2) *decide what to learn* — use these cases to deliberately form a set of learning goals to change its knowledge so that such a misunderstanding is not repeated on similar stories, and then (3) *strategy selection* — use nonlinear planning techniques to choose or construct some learning method by which it achieves these goals. These stages are detailed in Figure 1.

Previous publications have dealt with the blame assignment stage (Cox, 1993; Cox & Ram, 1992; Ram & Cox, 1994). This paper explores how learning goals are spawned when deciding what to learn (Section 2) and how these goals are satisfied in the strategy-selection phase (Section 3). A simpler system might forego explicit learning-goals altogether, and directly map a failure to a learning algorithm. The discussion of Section 3 explores, not only how learning goals are managed, but what leverage is gained over and above a direct mapping itself. The paper's conclusion (Section 4) specifies the relation between the Meta-AQUA system and traditional CBR approaches and discusses related systems, limitations with Meta-AQUA, and areas for future research.

```
0. Perform and Record Reasoning in Trace
1. Failure Detection on Reasoning Trace
2. If Failure Then
      Learn from Mistake:
        • 2a. Blame Assignment
              Compute index as characterization of failure
              Retrieve Meta-XP
              Apply Meta-XP to trace of reasoning
              If XP application is successful then
                 Check XP antecedents
                 If one or more nodes not believed then
                    Introspective questioning
                    GOTO step 0
                 Else GOTO step 0
        • 2 b. Create Learning Goals
              Compute tentative goal priorities
        • 2 c. Choose Learning Algorithm(s)
              Translate Meta-XP and goals to predicates
              Pass goals and Meta-XP to Nonlin
              Translate resultant plan into frames
        • 2 d. Apply Learning Algorithm(s)
              Interpret plan as partially ordered network of
                    actions such that primitive actions are
                    algorithm calls
3. Evaluate Learning (not implemented)
```

Figure 1: Meta-AQUA's learning algorithm

S1: A police dog sniffed at a passenger's luggage in the Atlanta airport terminal.

S2: The dog suddenly began to bark at the luggage.

S3: The authorities arrested the passenger, charging him with smuggling drugs.

S4: The dog barked because it detected two kilograms of marijuana in the luggage.

Figure 2: The drug-bust story

2 Deciding What to Learn

Learning goals represent what a system needs to know (Ram, 1991; 1993; Ram & Hunter, 1992; Ram & Leake, in press) and are spawned when deciding what to learn. Learning goals help guide the learning process by suggesting strategies that would allow the system to learn the required knowledge.[1] Given some failure of a reasoner, the task of the learning system is to adjust its knowledge so that such reasoning failures will not recur in similar situations.[2] The learner is therefore modeled as a planning system that spawns goals to achieve this overall task (Hunter, 1990; Ram & Hunter, 1992). The learner subsequently attempts to create plans resulting in desired new states of its background knowledge[3] that satisfy these goals. The overall aim is to turn reasoning failures into opportunities to learn and to improve the system's performance.

1. Learning goals also facilitate opportunistic learning (see Ram, 1991; 1993; Ram & Hunter, 1992), that is, if all information necessary for learning is not available at the time it is determined what is needed to be learned (e.g., when a question is posed), then a learning goal can be suspended, indexed in memory, and resumed at a later time when the information becomes available.

2. The learner could also adjust its circumstances in the physical world, such as placing items in a cupboard in the same place to aid memory retrieval. This paper, however, will not entertain such possibilities. See Hammond (1990) for an approach to such task interactions and associated learning.

3. The background knowledge includes more than simple domain knowledge. It can also contain knowledge such as metaknowledge, heuristic knowledge, associative knowledge, and knowledge of process.

Learning goals deal with the structure and content of knowledge, as well as the ways in which knowledge is organized in memory. Some learning goals seek to add, delete, generalize or specialize a given concept or procedure. Others deal with the ontology of the knowledge, i.e., with the kinds of categories that constitute particular concepts. Many learning goals are unary in that they take a single target as argument. For example, a *knowledge acquisition goal* seeks to determine a single piece of missing knowledge, such as the answer to a particular question. A *knowledge refinement goal* seeks a more specialized interpretation for a given concept in memory, whereas a *knowledge expansion goal* seeks a broader interpretation that explores connections with related concepts. Other learning goals are binary in nature since they take two arguments. A *knowledge differentiation goal* is a goal to determine a change in a body of knowledge such that two items are separated conceptually. In contrast, a *knowledge reconciliation goal* is one that seeks to merge two items that were mistakenly considered separate entities. Both expansion goals and reconciliation goals may include/spawn a *knowledge organization goal* that seeks to reorganize the existing knowledge so that it is made available to the reasoner at the appropriate time, as well as modify the structure or content of a concept itself. Such reorganization of knowledge affects the conditions under which a particular piece of knowledge is retrieved or the kinds of indexes associated with an item in memory.

A program called Meta-AQUA (Ram & Cox, 1994) was written to test our theory of understanding, explanation and learning. Given the drug-bust story of Figure 2, the system attempts to understand each sentence by incorporating it into its current story representation, explain any anomalous or interesting features of the story, and learn from any reasoning failures. Numerous inferences can be made from this story, many of which may be incorrect or incomplete, depending on the knowledge of the reader. Meta-AQUA's background knowledge includes general facts about dogs and sniffing, including the fact that dogs bark when threatened, but it has no knowledge of police dogs. It also knows cases of gun smuggling, but has never seen drug interdiction. The learning task in Meta-AQUA is to learn from failures, incrementally improving its ability to interpret new stories.

In the drug-bust story, sentence S1 produces no inferences other than that sniffing is a normal event in the life of a dog. However, S2 produces an anomaly because the system's definition of "bark" specifies that the object of a bark is animate. So the program (incorrectly) believes that dogs bark only when threatened by animate objects (see Figure 3 for the representation[4] produced by Meta-AQUA during blame assignment). Since luggage is inanimate, there is a contradiction, leading to an incorporation failure. This anomaly causes the understander to ask why the dog barked at an inanimate object. It is able to produce but one explanation: the luggage somehow threatened the dog.

S3 asserts an arrest scene which reminds Meta-AQUA of a past case of weapons smuggling by terrorists; however, the sentence generates no new inferences concerning the previous anomaly. Finally, S4 causes the question generated by S2, "Why did the dog bark at the luggage?" to be retrieved. Instead of revealing the anticipated threatening situation, S4 offers another hypothesis: "The dog detected drugs in the luggage."

The system characterizes the reasoning error as an expectation failure caused by the incorrect retrieval of a known explanation ("dogs bark when threatened by objects," erroneously assumed to be applicable), and a missing explanation ("the dog barked because it detected marijuana," the correct explanation in this case). During blame assignment, Meta-AQUA uses this characterization as an index to retrieve an abstract case called a Meta-XP (Ram & Cox, 1994) that is applied to a trace of the reasoning that produced the failure. The instantiation results in an explanation of its reasoning error, as shown in Figure 3. This composite meta-explanation consists of three parts: a Novel-Situation centered about "Retrieval Failure," an Erroneous-Association centered about "Expectation Failure" and an Incorrect-Domain-Knowledge centered about "Incorporation Failure."

Faced with the structure of the reasoning error produced by the blame-assignment phase, the learner determines the learning goals for the system. First, since the seemingly anomalous input (marked "Old Input" in Figure 3) has been incorporated in the story and later reinforced by the coherence of the story structure, and since no contradictions occurred as a result of this inference, the learner posts a knowledge reconciliation goal. The goal is to adjust the background knowledge so that neither dogs barking at animate objects nor dogs barking at inanimate objects will be considered anomalous by the understander. This learning goal is appropriate because even though one item is an

4. Attributes and relations are represented explicitly in Meta-AQUA and in this figure. For instance, the `ACTOR` attribute of an event Dog-bark.12 with the value Dog.4 is equivalent to the explicitly represented relation ACTOR.21 having a `domain` value of Dog-bark.12 and a `co-domain` value of Dog.4. In addition, all references to `TRUTH` attributes equal to `out` refer to the `domain` being out of the current set of beliefs. See Cox (1993) and Ram & Cox (1994) for further representational details. The "Internal Structures Window" of Figure 4 shows the top-level frame representation corresponding to Figure 3.

instantiated token (a particular dog barked at a specific inanimate object), while the other is a type definition (concept specifying that dogs generally bark at animate objects), they are similar enough to each other to be reconcilable.

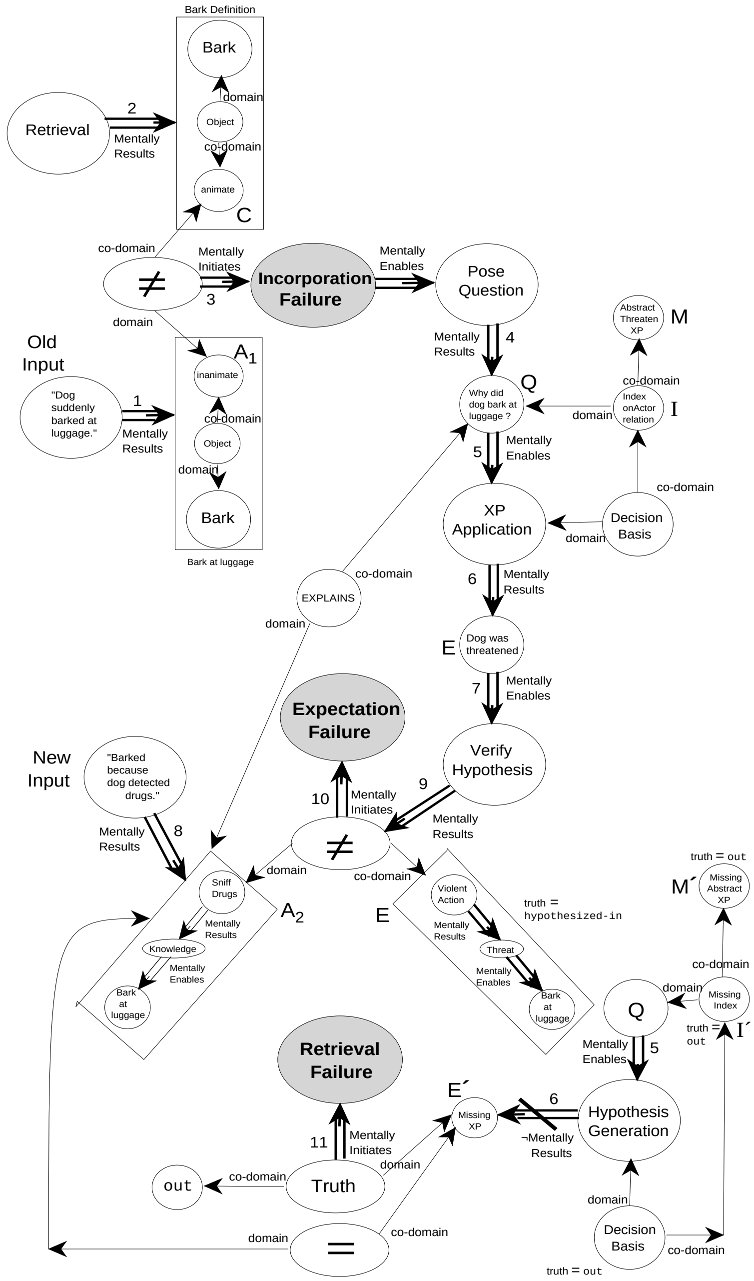

Figure 3: Instantiated composite meta-explanation

Secondly, given that an expectation failure triggered the learning, and (from the blame assignment phase) given that the failure resulted from the interaction of misindexed knowledge and a novel situation, Meta-AQUA posts a goal to differentiate between the two explanations for why the dog barked (nodes M and M′ in Figure 3). Since the conflicting explanations are significantly different (for example, they do not share the same predicate, i.e., detect versus threaten), a knowledge-differentiation goal is licensed, rather than a goal to reconcile the two types of explanations. The differentiation goal is achieved if the system can retrieve proper explanations given various situations. The original misunderstanding of the story occurred, not because the explanation that dogs bark when threatened is incorrect in general, but rather because the system did not know the proper conditions under which this explanation applies.

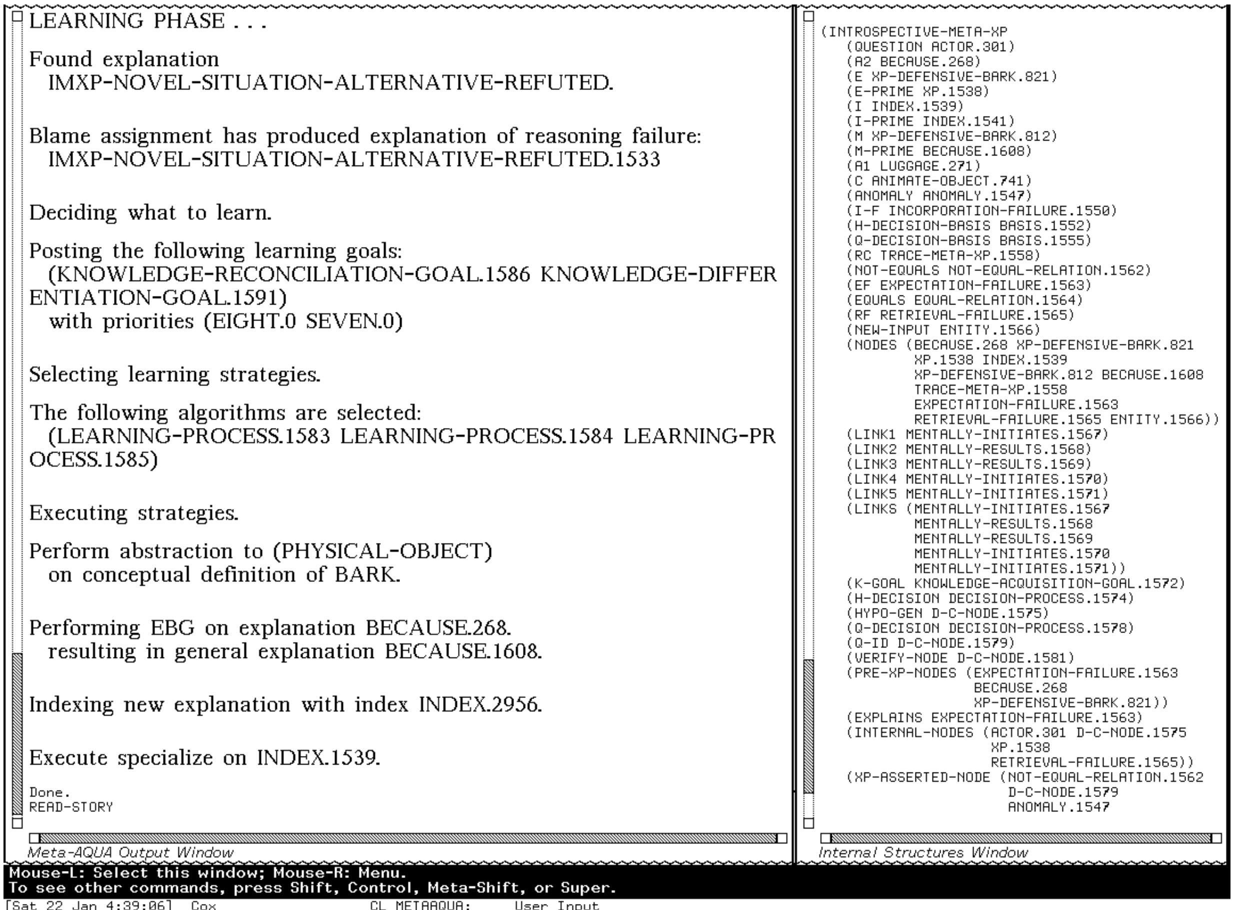

Figure 4: Meta-AQUA output and frame representation of the meta-explanation used in example story

In addition to posting these two learning goals, Meta-AQUA places a tentative ordering on their execution (see Figure 4 for program output when deciding what to learn). With no other specific knowledge concerning their respective relations, a good default heuristic is to order them by temporal order of the failures involved in the original reasoning trace. The reason this may be useful is that if it is determined that the first failure was indeed not an error but a misunderstanding or was caused by faulty input, then the reasoning that followed from the first failure (or other assumptions depending on the nature of the first failure that led to the second) may have contributed to the cause of the second. Thus, learning acquired from the first failure may show that the subsequent reasoning was irrelevant, or it may yield more information to be used on the second goal. Therefore, the decide-what-to-learn stage outputs the knowledge-reconciliation goal with priority over the knowledge-differentiation goal.

3 Strategy Selection and Combination

In the strategy-selection stage, Meta-AQUA constructs a learning plan to realize the learning goals posted by the previous stage (see Figure 4 for program output during strategy selection phase). This entails not only choosing the algorithms and operators to achieve the learning goals, but perhaps also spawning new subgoals. To help the system create a plan for a learning goal with two arguments, the following types of questions can be posed about the reasoning chain. For example, with the knowledge-differentiation goal, the focus starts at the error that triggered the introspective-explanation episode, that is, at the expectation failure. Given that the reasoner expected explanation E to be correct, but later decides that A_2 is the actual explanation, the system needs to determine:

- Was the actual occurrence, A_2, foreseeable?
- If so, was A_2 considered?
- Is there still a possibility that A_2 is incorrect?

- Is there still a possibility that E is correct?
- How confident is the system in A_2 or any of the input associated with establishing A_2?
- How much experience does the system have with A_2 and E's abstract progenitors?

The answers to these questions enable the system to choose learning algorithms, strategies, or operators. For example, since the explanation provided by the story (A_2) provides more coherence to the understanding of the story, the system assumes that there is no error in the input. However, because the system has no prior experience with the instance (and thus the system neither foresaw nor considered the explanation), the learner posts another goal to expand the concept, thus producing M'. Explanation-based generalization (EBG) (DeJong & Mooney, 1986; Mitchell, Keller & Kedar-Cabelli, 1986) can then be selected as an appropriate learning algorithm.

A more difficult problem is to differentiate the applicability conditions for the two generalized explanations (M', the one produced by generalizing the detection explanation, A_2, and M, the original abstract pattern that produced the initial threaten explanation, E) by modifying the indexes (I' and I) with which the system retrieves those explanations. If the two problems of erroneous association and novel situation were to be treated independently, rather than as a problem of interaction, then an indexing algorithm would not be able to ensure that the two explanations would remain distinct in the future. That is, if the learner simply detects a novel situation and automatically generalizes it, then indexes it by the salient or causal features in the explanation, and if the learner independently detects an erroneous retrieval, and re-indexes it so that the same context will not retrieve it in the future, then there is no guarantee that the resultant indexes will be mutually exclusive. Instead, the system must re-index E with respect to A_2, not simply with respect to the condition with which E was retrieved. Therefore, a direct mapping from blame assignment to strategy selection without the mediation of learning goals is problematic.

The problems to be solved, then, are determining the difference between A_2 and E, and, in the light of such differences, computing the minimal specialization of the index of E and the maximally general index of A_2 so they will be retrieved separately in the future. In the case of the story above, the problem is somewhat simplified. The difference is that retrieval based on the actor relation of barking actions (dogs) is too general. The threaten explanation applies when dogs bark at animate objects, while the detection explanation is appropriate when dogs bark at containers.

The knowledge-reconciliation goal between the conceptual definition of dog-barking being limited to animate objects and the fact that a particular dog barked at a piece of luggage can be thought of as a simple request for similarity-based learning (SBL) or inductive learning (for example, UNIMEM's SBL algorithm in Lebowitz, 1987, or abstraction transmutation as in Michalski, 1994). The system is simply adding an additional positive example to the instances seen. An incremental algorithm is required because this instance has been discovered after an initial concept has been established some time in the past.

An interesting interaction can occur, however, if the system waits for the result of the EBG algorithm required by the knowledge-expansion subgoal spawned by the knowledge-differentiation goal discussed above. The algorithm will generalize the explanation (that this particular dog barked at a particular piece of luggage because it detected marijuana) to a broader explanation (that dogs in general may bark at any container when they detect contraband). Thus, the example provided to the inductive algorithm can be more widely interpreted, perhaps allowing its inductive bias to generalize the constraint, C, on the object of dog-barking to `physical-object` (the exhaustive case of `animate-object` and `inanimate-object`), whereas a single instance of a particular breed of dog barking at a specific brand of luggage, A_1, may limit the inductive inference if no additional domain knowledge is available.

Unfortunately, however, because the EBG algorithm uses the representation of the dog-bark definition, and the inductive algorithm changes this definition, the induction must occur first. Thus, the learner cannot take advantage of the opportunity cited in the previous paragraph. One important implication of this point is that in systems which plan to learn, if the reasoner does not anticipate this second interaction (thus placing EBG before the induction), the system must be able to perform dynamic backtracking on its decisions.

To notice these types of interactions, however, requires a least-commitment approach such as that used in a non-linear hierarchical planner like Nonlin (Ghosh, Hendler, Kambhampati, & Kettler, 1992; Tate, 1976). Likewise, the system must detect any dependency relationships so that goal violations can be avoided. For example, when the definition of dog-barking is modified by generalizing its constraint of what dogs bark at to `physical-object` from `animate-object`, any indexing based on the modified attribute must occur after this modification, rather than before it, to avoid indexing with obsolete conceptual knowledge.[5]

Figure 5 shows a learning-operator definition for the indexing strategy that manages mutual indexing between two concepts. The operator schema determines that both items must be independently indexed before they are indexed with respect to each other. The action schema has filter conditions that apply when both are indexed and both are XPs. An unsupervised condition specifies that if there exists a change in the explained action, then it must occur before the execution of this schema. That is, a linearization must be performed on external goals to reorder any other schema that may establish the change. It says in effect that we want all attributes of the target concept to be stable before it operates on the concept; no other operator can change an attribute in order for the changes performed by indexing to be unaffected. Note that the action schema of abstraction in Figure 6 has an effect that includes such a change to its addlist. Therefore, if both schemas are being instantiated, Nonlin will automatically order the abstraction before the indexing. A similar unsupervised condition prevents generalization from occurring before the concept is stable.

```
(opschema mutual-index-op
       :todo   (index-wrt-item ?x ?y)
       :expansion (
              (step1 :goal (indexed ?x))
              (step2 :goal (indexed ?y))
              (step3 :action
                    (index-dual-items ?x ?y)))
       :orderings(
              (step1 -> step3)
              (step2 -> step3))
       :conditions (
              (:precond (indexed ?x)
               :at step3 :from step1)
              (:precond (indexed ?y)
               :at step3 :from step2)
              (:use-when (not (equal ?x ?y))
                    :at step1))
       :effects   ()
       :variables (?x ?y))
```

```
(actschema do-mutual-xp-indexing
       :todo   (index-dual-items ?x ?y)
       :expansion ( (step1 :primitive
                    (perform-mutual-indexing ?x ?y)))
       :conditions (
              (:use-when (indexed ?x) :at step1)
              (:use-when (indexed ?y) :at step1)
              (:use-when (isa xp ?x) :at step1)
              (:use-only-for-query
                    (explains ?explains-node ?x)
               :at step1)
              (:use-only-for-query
                    (domain ?explains-node
                     ?explained-action)
               :at step1)
              (:unsuperv (changed true ?explained-action)
               :at step1)
                   )
       :effects  (
              (step1 :assert (indexed-wrt ?x ?y))
              (step1 :assert (indexed-wrt ?y ?x)))
       :variables (?x ?y ?explains-node ?explained-action))
```

Figure 5: Mutual-indexing schemas

```
(actschema do-abstraction-change
       :todo   (abstracted ?r1 ?r2)
       :expansion ( (step1 :primitive (perform-abstraction ?r1 ?r2)))
       :conditions (
              (:use-when (isa relation ?r1) :at step1)
              (:use-when (isa relation ?r2) :at step1)
              (:use-when (relation ?r1 ?r1-type) :at step1)
              (:use-when (relation ?r2 ?r2-type) :at step1)
              (:use-only-for-query (domain ?r1 ?r1-domain) :at step1)
              (:use-only-for-query (co-domain ?r1 ?c) :at step1)
              (:use-only-for-query (co-domain ?r2 ?a) :at step1)
              (:use-only-for-query (parent-of ?c ?c-parent) :at step1)
              (:use-only-for-query (parent-of ?a ?a-parent) :at step1)
              (:use-when (equal ?r1-type ?r2-type) :at step1)
              (:use-when (equal ?c-parent ?a-parent) :at step1))
       :effects  (
              (step1 :assert (co-domain ?r1 ?c-parent))
              (step1 :assert (changed true ?r1-domain))
              (step1 :delete (co-domain ?r1 ?c))
              (step1 :delete (changed false ?r1-domain)))
       :variables (?r1 ?r2 ?r1-type ?r2-type ?r1-domain ?c ?a ?c-parent ?a-parent))
```

Figure 6: Abstraction schema

Therefore, the final learning plan Meta-AQUA constructs is (1) perform an abstraction transmutation on the new example of dog barking (realizing that dogs bark at containers); (2) perform EBG on the new explanation (producing a generalized version); (3) index the generalized XP in isolation; and finally, (4) use the new concept definition to mutually differentiate and index the two generalized explanations of why dogs bark. A subsequent story, such that a police officer and a canine enter a suspect's house, the dog barks at a garbage pail, and the suspect is arrested for possession of some marijuana found in the pail, causes no anomaly. Indeed, Meta-AQUA expects some type of contraband to be found in the container after it reads that the dog barked, but before it is told of its existence in the story. Thus, the learning improves both understanding and prediction.

5. This result supersedes the conjecture by Ram & Hunter (1992) that, unlike standard planning techniques, interactions and dependencies do not occur with learning goals.

4 Conclusions

Although Meta-AQUA is firmly in the CBR tradition, our approach diverges from it somewhat. Three elements traditionally characterize CBR. First, CBR usually processes instances or concrete episodic cases. However, some systems emphasize the integration of generalized knowledge and cases (e.g., Aamodt, 1993), and moreover, like Meta-AQUA, some CBR systems actually process abstract cases, including XPs (see Schank, Kass & Riesbeck, 1994). Secondly, CBR emphasizes the role of memory retrieval of past examples, rather than reasoning from first principles. This focus has led to research on indexing vocabulary and case adaptation. However, Meta-AQUA is a hybrid system that combines the CBR of the first two learning phases with the nonlinear planning of the third. Finally, traditional CBR systems stress goal-directed activity to focus both processing and learning (Kolodner, 1993; Ram & Hunter, 1992; Schank, 1982). Our approach to learning is also goal-directed, but in a very different style. Meta-AQUA is the first CBR system to specifically plan in the knowledge space using goals that specify changes in that space. Unlike INVESTIGATOR (Hunter, 1990), which creates plans in the external world to achieve learning goals (e.g., access a database to answer a question), Meta-AQUA's plans operate on the internal word of the system's background knowledge. Although many computational systems use a reflective reasoning approach (e.g., Collins, Birnbaum, Krulwich, & Freed, 1993; Fox & Leake, 1994; Oehlmann, Edwards, & Sleeman, 1994; Plaza & Arcos, 1993; Stroulia & Goel, in press), and a few have used the planning metaphor in learning (Hunter, 1990; Quilici, in press; Ram & Hunter, 1992; Ram & Leake, in press; Redmond, 1992), none of these systems have applied the planning metaphor as strictly as Meta-AQUA; none execute a planner like Nonlin on its own knowledge.

A number of advantages accrue from the mediation of learning through satisfaction of learning goals. First, learning goals decouple the many-to-many relationship between failures and algorithm. Secondly, an opportunistic approach to solving learning problems can be achieved by suspending the goals and resuming their pursuit at a time when satisfaction is more likely. Thirdly, learning goals allow chaining, composition, and optimization of the means by which learning goals are achieved. Fourthly, because nonlinear plans allows parallelism, learning algorithms may be executed concurrently. Finally, the use of learning goals allows detection of dependency relationships so that goal violations can be avoided.

Future research is directed toward incorporating more learning strategies. One of the weak points of the current system is that it reasons during learning at a macro-level. Meta-AQUA recognizes the functional difference between generalization and specialization and therefore can choose an appropriate algorithm based on which algorithm is most appropriate. For example, it cannot currently select between competing algorithms that both perform generalization. Meta-AQUA does not reason at the micro-level, as do systems that address the selective-superiority problem[6] in inductive learning (see, for instance, Brodley, 1993; Provost & Buchanan, 1992; Schaffer, 1993), although the scope of learning problems solved by Meta-AQUA is greater than these other systems.

Another limitation of the Meta-AQUA implementation is that learning self-evaluation (step 3 of Figure 1) does not exist. Thus, Meta-AQUA cannot cross-validate or compare various successful algorithms, nor can it currently judge when learning fails, and another algorithm must be chosen. Just as it detects, explains, repairs and learns from reasoning failures, an interesting line of future research is to allow Meta-AQUA to reason about its own learning.

To perform multistrategy learning, a CBR system must consider a number of factors that are not significant in isolated learning systems. In particular, a system must be able to handle insufficient resources and knowledge and manage dependency relations between learning algorithms at run-time. Many alternative solutions and interactions may occur, even when reasoning about simple situations. Treating the learner as a planner is a principled way to confront these difficulties. Many of the techniques and results from the planning literature can be appropriated in case-based systems to provide a better level of robustness and coverage in situations where many types of failure may occur. The aim is to transform these failures into opportunities to learn and improve the system's overall performance.

Acknowledgments

This research is supported by AFOSR under contract #F49620-94-1-0092 and by the Georgia Institute of Technology.

6. Empirical results suggest that various inductive algorithms are better at classifying specific classes or particular distributions of data than others. Each algorithm is good at some but not all learning tasks. The selective superiority problem is to choose the most appropriate inductive algorithm, given a particular set of data (Brodley, 1993).

References

Aamodt, A. (1993). Explanation-driven retrieval, reuse and learning of cases. In M. M. Richter, S. Wess, K.-D. Althoff, & F. Maurer (Eds.), *Proc. of the First European Workshop on Case-Based Reasoning,* Vol. 2 (pp. 279-284). SEKI Tech. Rep. No. SR-93-12 (SFB 314). Univ. of Kaiserslautern, Germany.

Brodley, C. E. (1993). Addressing the selective superiority problem: Automatic algorithm / model class selection. *Machine Learning: Proc. of the Tenth International Conference* (pp. 17-24). San Mateo, CA: Morgan Kaufmann.

Collins, G., Birnbaum, L. Krulwich, B. & Freed, M. (1993). The role of self-models in learning to plan. In A. L. Meyrowitz & S. Chipman (Eds.), *Foundations of knowledge acquisition* (pp. 117-143). Boston: Kluwer Academic Publishers.

Cox, M. T. (1993). *Introspective multistrategy learning* (CogSci. Tech. Rep. No. 2). Atlanta: Georgia Tech, College of Computing.

Cox, M. T., & Ram, A. (1992). Multistrategy learning with introspective meta-explanations. D. Sleeman & P. Edwards (Eds.), *Machine Learning: Proceedings of the Ninth International Conf.* (pp. 123-128). San Mateo, CA: Morgan Kaufmann.

DeJong, G., & Mooney, R. (1986). Explanation-based learning: An alternative view, *Machine Learning*, *1*(2), 145-176.

Fox, S., and Leake, D. (1994). Using Introspective Reasoning to Guide Index Refinement in Case-Based Reasoning. In *Proceedings of the Sixteenth Annual Conferences of the Cognitive Science Society* (pp. 324-329). Hillsdale, NJ: Lawrence Erlbaum.

Ghosh, S., Hendler, J., Kambhampati, S., and Kettler B. (1992). *UM Nonlin* [a Common Lisp implementation of A. Tate's Nonlin planner]. Maintained at the Dept. of Computer Science, University of Maryland, College Park, MD. Available by anonymous ftp from cs.umd.edu in directory /pub/nonlin.

Hammond, K. J. (1990). Learning and enforcement: Stabilizing environments to facilitate activity. B. W. Porter & R.Mooney (Eds.), *Machine Learning: Proc. of the Seventh International Conf.* (pp. 204-210). San Mateo, CA: Morgan Kaufmann.

Hunter, L. E. (1990). Planning to learn. *Proc. of Twelfth Annual Conference of the Cognitive Science Society* (pp. 261-276). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kolodner, J. L. (1993). *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.

Lebowitz, M. (1987). Experiments with incremental concept formation: UMIMEM. *Machine Learning*, *2*, 103-138.

Michalski, R. S. (1994). Inferential theory of learning: Developing foundations for multistrategy learning. In R. S. Michalski & G. Tecuci (Eds.), *Machine learning: A multistrategy approach IV* (pp. 3-61). San Francisco: Morgan Kaufmann Publishers.

Mitchell, T., Keller, R., & Kedar-Cabelli, S. (1986). Explanation-based generalization: A unifying view, *Machine Learning, 1*(1), pp. 47-80.

Oehlmann, R., Edwards, P., and Sleeman, D. (1994). Changing the viewpoint: Re-indexing by introspective questioning. In *Proc. of the 16th Annual Conf. of the Cognitive Science Society* (pp. 675-680). Hillsdale, NJ: Lawrence Erlbaum Associates.

Plaza, E., & Arcos, J. L. (1993). Reflection and analogy in memory-based learning. In R. S. Michalski & G. Tecuci (Eds.), *Proc. of the Second Intl. Workshop on Multistrategy Learning* (pp. 42-49). Center for Art. Intel., George Mason Univ., Fairfax, VA.

Provost, F. J., & Buchanan, B. G. (1992). Inductive policy. *Proc. of the Tenth National Conference on Artificial Intelligence* (pp. 255-261). Menlo Park, CA: AAAI Press.

Quilici, A. (in press). Toward automatic acquisition of an advisory system's knowledge base. *Applied Intelligence.*

Ram, A. (1991). A theory of questions and question asking. *The Journal of the Learning Sciences*, *1*(3&4), 273-318.

Ram, A. (1993). Indexing, elaboration and refinement: Incremental learning of explanatory cases. *Machine Learning*, *10*(3), 201-248.

Ram, A., & Cox, M. T. (1994). Introspective reasoning using meta-explanations for multistrategy learning. In R. S. Michalski & G. Tecuci (Eds.), *Machine learning: A multistrategy approach IV* (pp. 349-377). San Francisco: Morgan Kaufmann.

Ram, A., & Hunter, L. (1992). The use of explicit goals for knowledge to guide inference and learning. *Applied Intelligence, 2*(1), 47-73.

Ram, A., & Leake, D. (in press). Learning, goals, and learning goals. In A. Ram & D. Leake (Eds.), *Goal-driven learning*. Cambridge, MA: MIT Press/Bradford Books.

Redmond, M. A. (1992). *Learning by observing and understanding expert problem solving* (Tech. Rep. GIT-CC-92/43). Doctoral dissertation, Atlanta: Georgia Institute of Technology, College of Computing.

Schaffer, C. (1993). Selecting a classification method by cross-validation. *Machine Learning*, *13*(1), 135-143.

Schank, R. C. (1982). *Dynamic memory: A theory of reminding and learning in computers and people*. Cambridge, UK: Cambridge University Press.

Schank, R. C., Kass, A., & Riesbeck, C. K. (1994). *Inside case-based explanation*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Stroulia, E., & Goel, A. (in press). Functional representation and reasoning for reflective systems. In *Applied Art. Intelligence*.

Sussman, G. J. (1975). *A computer model of skill acquisition*. New York: American Elsevier.

Tate, A. (1976). *Project planning using a hierarchic non-linear planner.* Tech. Rep. #25, Dept. of Art. Intel., Univ. of Edinburgh.