An Empirical Study of Computational Introspection: Evaluating Introspective Multistrategy Learning in the Meta-AQUA System

Michael T. Cox
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213-3891
mcox+@cs.cmu.edu

Abstract

The theory of introspective multistrategy learning proposes that three transformations must occur to learn effectively from a performance failure in an intelligent system: Blame assignment, deciding what to learn, and learning-strategy construction. The Meta-AQUA system is a multistrategy learner that operates in the domain of story-understanding failures and is designed to evaluate this learning approach. We discuss experimental results supporting the hypothesis that introspection facilitates learning in a multistrategy environment. In an empirical study, Meta-AQUA performed significantly better with a fully introspective mode than with a reflexive mode in which learning goals were ablated. In particular, the results lead to the conclusion that the process which posts learning goals (deciding what to learn) is a necessary transformation if negative interactions between learning methods are to be avoided and if learning is to remain effective. Moreover, we show that because learning algorithms can negatively interact, the arbitrary ordering of learning methods can actually lead to worse system performance than no learning at all. The goals of this research are to better understand the interactions between learning algorithms, to determine the role of introspective mechanisms when integrating them, and to more firmly establish the conditions under which such an approach is warranted (and those under which it is not).

Introduction

From the very early days of AI, researchers have been concerned with the issues of machine self-knowledge and introspective capabilities (e.g., McCarthy 1959; Minsky 1954/1965), yet few have quantitatively evaluated the trade-offs involved with much care or investigated the nature of the role introspection assumes when learning. The research presented here uses computational introspection to assist in the choice and sequencing of learning algorithms within a multistrategy framework. Yet open questions exist as to whether introspection is worth the computational overhead and in exactly what ways it facilitates the learning process. This paper begins to investigate these research questions empirically.

The theory of *introspective multistrategy learning* (IML) proposes that three transformations must occur to learn effectively from performance failures. First, given a trace of the performance failure, a learner must perform *blame assignment* by mapping the symptoms of the failure to a causal explanation of the failure. Secondly, the learner must use this explanation to *decide what to learn* by posting explicit learning goals to achieve desired changes in its background knowledge. Thirdly, the learner can use these goals for *learning-strategy construction* by treating the learning task as a nonlinear planning problem. That is, the learner constructs a partially-ordered plan to repair the background knowledge by sequencing calls to standard learning algorithms. The Meta-AQUA system (Cox 1996; Ram & Cox 1994) is a multistrategy learner that operates in the domain of story understanding failures and is designed to evaluate this learning approach.

Section 2 briefly presents the Meta-AQUA system by describing the story generation module with which experimental trials are generated and by providing a brief explanation of the performance and learning tasks. Section 3 provides a computational evaluation of the hypothesis that introspection facilitates learning using data obtained from the Meta-AQUA system. Section 4 summarizes the results and concludes with a short discussion.

Meta-AQUA

Meta-AQUA is a learning system that chooses and combines multiple learning methods from a toolbox of algorithms in order to repair faulty components responsible for failures encountered during the system's performance task. The system architecture and flow of information within Meta-AQUA is shown in Figure 1. The problem generation module outputs a story to the story-understanding performance system with the initial goal to understand the input. The performance module uses schemas from its background knowledge (BK) to explain the story and to build a representation for it in its foreground knowledge (FK). If this task fails, then a trace of the reasoning that preceded the failure is passed to the learning subsystem. A case-based reasoning (CBR) (Kolodner, 1993) subsystem within the learner uses past cases of introspective reasoning from the BK to explain the comprehension failure and to generate a set of learning goals. These goals, along with the trace, are then passed to a nonlinear planner. The planner subsequently builds a learning strategy from its toolbox of learning methods. The learning plan is passed to an execution system that examines and changes items in the BK. These changes enable improved future performance.

The above conceptualization of learning is consistent with both Michalski's (1994) Inferential Learning Theory that decomposes a learning task into an input, the BK, and a learning goal and Carbonell (1986) and Veloso's (1992) emphasis on reasoning from a trace of the derivation of a solution rather than from solutions themselves. Although the algorithms and knowledge structures used by Meta-AQUA have been reported elsewhere (e.g., Cox 1994, 1996; Cox & Ram 1995; Ram & Cox 1994; Ram, Cox, & Narayanan 1995), this section outlines the system in order to provide a context for understanding the evaluation.

The Input: Elvis World and Tale-Spin

To support large data collection, the Tale-Spin story generation program[1] provides a potentially infinite number of input variations that test Meta-AQUA's ability to learn from explanation failure. Given a main character and a problem, Tale-Spin simulates the actions that would be necessary for the character to achieve goals stemming from the problem. For example if a character is bored, Tale-Spin assigns the character an initial goal to remove the state of boredom. The character can achieve the goal by convincing a friend to play, finding a ball, going outside, and then batting the ball back and forth (see Figure 2). For each event in the story, the generator adds any associated causal results. These results change the world and enable further actions by characters in

[1]Tale-Spin (Meehan 1981) was obtained from the UC Irvine repository. Pazzani (1994) used it to evaluate the OCCAM multistrategy learning system.

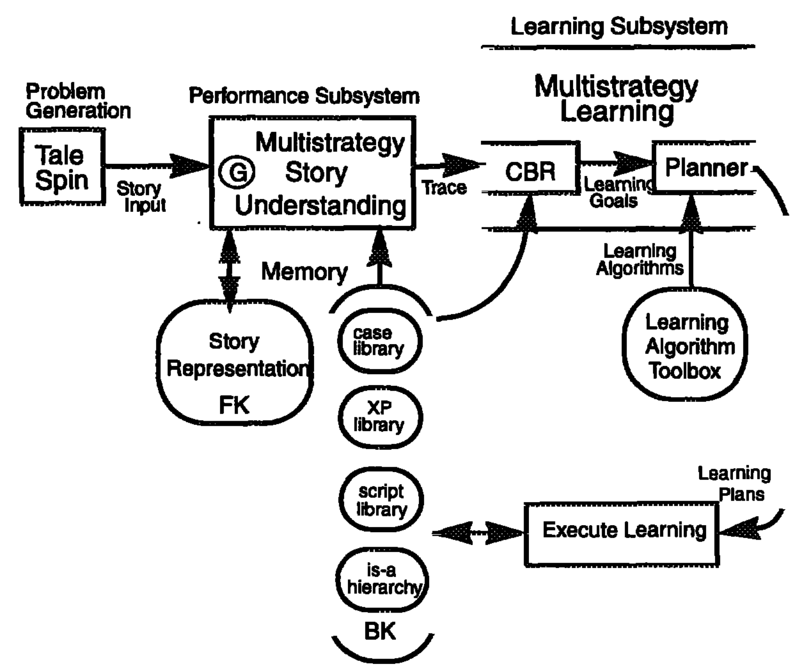

Figure 1. Detailed Meta-AQUA system architecture

the story. For example, the act of getting the ball and going outside enables the hitting of the ball which results in the ball's movement between the characters. In turn, these actions remove the boredom. The story terminates when the goals and subgoals of the main character have been achieved or when all possible plans to achieve them have been exhausted.

```
Elvis asked Lynn, "Would you push the ball2 to me away from you?" Lynn went to the garage. She picked up the ball2. She had the ball2. She went to outside. He went to outside. He played with the ball2. She hit the ball2. She hit the ball2 because she wanted to move the ball2 to him. He hit the ball2. He hit the ball2 because he wanted to move the ball2 to her. He played with the ball2 because he didn't want to be bored.

--- The End ---
```

Figure 2. Sample Elvis World story

Among the changes to Tale-Spin, we added a musician named Elvis and a police officer to the cast of characters. Elvis is temporarily boarding with Mom, Dad and their daughter Lynn, whereas the officer occasionally visits the house, presumably because of neighborhood complaints of loud music and raucous behavior. Furthermore, the police officer often (but not always) brings a drug-detection dog along with him. We also removed Karen from the cast of main characters available as a protagonist and the state of hunger from the possible initial problem states. Thus, Tale-Spin now generates a more uniform distribution of situations.

We also added two new problem types to the original problems of thirst and boredom. Characters may now be *jonesing*[2] for drugs. In Elvis' case, he sometimes smokes marijuana to relieve his jones, whereas Dad occasionally smokes a pipe with tobacco. Lynn has also been given a tobacco habit. The police officer has the problem of being *concerned* about the law. The police officer's state of being concerned is relieved if he can either locate contraband or arrest criminals.[3] We also reprogrammed Tale-Spin to hide the marijuana during story initialization in different locations (e.g, in the cupboard, refrigerator, and under the carpet), so the officer's task varies depending on entry conditions (i.e., at what point in the story the officer arrives on the scene and whether the dog accompanies him), the initial location of the pot, and the actions of the characters in the story.

Finally, to facilitate the performance task, we modified the Tale-Spin program so it generates explanations of key events in the stories. The resolution of all anomalies are thereby incorporated within each story. For example, Tale-Spin includes a reason why Lynn strikes the ball in the story above because it knows that Meta-AQUA will find the action anomalous and thus try to explain it. Although in an ideal implementation, the understanding process should be able to make independent inferences that confirm explanations of the input, Meta-AQUA depends on the story to provide explanations for this confirmation. The implementation concentrates on the learning task rather than the understanding task.

The Performance and Learning Tasks: Story Understanding, Explanation, and Repair

The Meta-AQUA system learns about drug-smuggling and sports activities, given its prior experience with stories about terrorists and its general knowledge of physical causality. The systems' performance task is to "understand" stories by building causal explanations that link the individual events into a coherent whole. The performance sub-system uses a multistrategy approach to understanding. Thus, the top-level goal is to choose a comprehension method (e.g., script processing, case-based reasoning, or explanation pattern application) by which it can understand the input. When an anomalous or otherwise interesting input is detected, the system builds an explanation of the event, incorporating it into the preexisting model of the story in the FK.

[2]In the vernacular, a "jones" is a drug habit accompanied by withdrawal symptoms. The verb "to jones" is to be going through a state of withdrawal.

[3]Unlike the UC Irvine version of Tale-Spin in which characters and their goals did not interact, we modified the program so that the police officer is a competing character with his own problem and goal. Because the officer confiscates the marijuana when found and arrests Elvis, such events may preempt the enabling conditions of actions Elvis had planned to perform. For instance, if Elvis is thirsty but the officer arrests him, this condition restricts his freedom of movement so that he cannot go to the faucet for water. Therefore, the story can end with Elvis still having the problem with which he began (i.e, thirst).

In the story from Figure 2 for example, Meta-AQUA finds it unusual for a person to strike a ball because its conceptual definition of the "hit" predicate constrains the object attribute to animate objects. It tries to explain the action by hypothesizing that Lynn tried to hurt the ball (an abstract explanation pattern, or XP, retrieved from the BK instantiates this explanation). In the following, sentence, however, the story specifies an alternate explanation (i.e., the hit action is intended to move the ball to the opposing person). This input causes an expectation failure because the system had expected one explanation to be true, but another proved true instead.

When the Meta-AQUA system detects an explanation failure, the performance module passes a trace of the reasoning to the learning subsystem. At this time, the learner needs to explain why the failure occurred (assign blame) by applying an introspective explanation to the trace. A meta-explanation (Meta-XP) is retrieved using the failure symptom as a probe into memory. Meta-AQUA instantiates the retrieved meta-explanation and binds it to the trace of reasoning that preceded the failure. The resulting structure is then checked for applicability. If the Meta-XP does not apply correctly, then another probe is attempted. An accepted Meta-XP either provides a set of learning goals (determines what to learn) that are designed to modify the system's BK or generates additional questions to be posed about the failure. Once a set of learning goals are posted, they are passed to the nonlinear planner for building a learning plan (strategy construction).

Figure 3 lists the major state transitions that the three learning processes produce. The learning plan is fully ordered to avoid interactions. For example, the abstraction step must precede the other steps. A knowledge dependency exists between the changes on the hit concept as a result of the abstraction and the use of this concept by both generalization and the indexing.[4] After the learning is executed and control returns to sentence processing, subsequent sentences concerning the hit predicate causes no anomaly. Instead, Meta-AQUA predicts the proper explanation.[5]

Symptoms:
- Contradiction between input and background knowledge
- Contradiction between expected explanation and actual explanation

Faults:
- Incorrect domain knowledge
- Novel situation
- Erroneous association

Learning Goals:
- Reconcile input with conceptual definition
- Differentiate two explanations

Learning Plan:
- Abstraction on concept of hit
- Generalization on hit explanation
- Index new explanation
- Mutually re-index two explanations

Figure 3. Learning from Explanation Failure

Computational Evaluation

This section presents the results of computational studies performed with Meta-AQUA to test the hypothesis that introspection facilitates learning. The methodology below not only tests our hypothesis, but also it more directly supports the position that a loose coupling of blame-assignment and repair (via learning goals) is preferred to a tight coupling approach. But perhaps more importantly, this methodology also scrutinizes the claim that the second phase of learning, deciding what to learn, is *necessary* for effective learning. IML theory is the only learning theory that makes such a strong claim. Few computational systems other than Meta-AQUA include an explicit calculation of a goal to learn and then use that goal to influence learning. Converging with the arguments and hand-coded examples from previous research that favor this position (e.g., Cox 1994; Cox & Ram 1995),

[4]During mutual re-indexing, the explanations are differentiated based on the object attribute-value of the hit. However, the abstraction transmutation changes this attribute. The generalization method applied to the new explanation also uses this attribute. See Cox & Ram (1995) for a more complete analysis.

[5]As pointed out by an anonymous reviewer, it would be nice for the system to use ontological knowledge to infer that the inanimate objects cannot feel pain. At the current time, however, the system possess neither the bias to make a proper inductive leap during learning nor the prerequisite knowledge to make the inference. Indeed, the system has but a primitive causal understanding of the mechanics of pain.

this paper presents quantitative evidence that supports the utility of this stage.

The Hypothesis

Generally, we claim that introspection facilitates learning. More specifically, we assert that the rate of improvement in story understanding with learning goals exceeds that of story understanding without learning goals holding all other factors constant. Our approach is to perform a kind of ablation study. Surgically removing the learning goals eliminates part of the system's mechanism responsible for introspection. The intention of this manipulation is to show different empirical learning curves with and without introspection as a function of the number of inputs.

Introspective learning is a computational process with the decomposition as shown in the upper portion of Figure 4. *Fully introspective multistrategy learning* consists of examining one's own reasoning to explain where the reasoning fails. It consists further of knowing enough about the self and one's own knowledge that the reasoner can explicitly decide what needs to be learned. Introspection amounts to performing blame assignment and subsequently posting explicit goals to learn. Learning amounts to the construction of a learning plan designed to change the reasoner's knowledge and thereby to achieve the learning goals.

Removing the goals from the introspective process above, leaves a more reflexive activity we call *semi-introspective multistrategy learning*[6] (see the lower portion of Figure 4). Instead of using the explanation of failure created during the blame-assignment phase to post a set of learning goals that then direct the construction of a learning plan, the explanation can directly determine the choice of repair methods. System performance under both conditions can then be compared with Meta-AQUA under a no learning situation.

Independent and Dependent Variables

Learning rates relative to a baseline no-learning condition are compared between the fully introspective and a semi-introspective version of Meta-AQUA. The independent variable that effects this change is the presence and influence of learning goals. The first experimental condition is referred to as the learning goal (LG) condition, and represents Meta-AQUA as described in Ram & Cox (1994). Under the LG condition, the system builds a learning strategy. This construction is guided by the learning goals spawned by the Meta-XPs that explain the failure. Hence, this condition represents a loose coupling approach between fault (failure cause) and repair (learning).

The second condition is called the random learning (RL) condition. Given the explanation of the causes of failure the system can directly assign calls to particular learning algorithms for each fault. The construction of the learning plan is then performed by a random ordering of these function calls, rather than by non-linear planning to achieve the learning goals. The RL condition represents a tight coupling

[6]It is semi-introspective because, although part of the introspective process has been removed, the introspective mechanics of blame-assignment remain. Future research remains to test the performance with blame assignment removed and learning goals present.

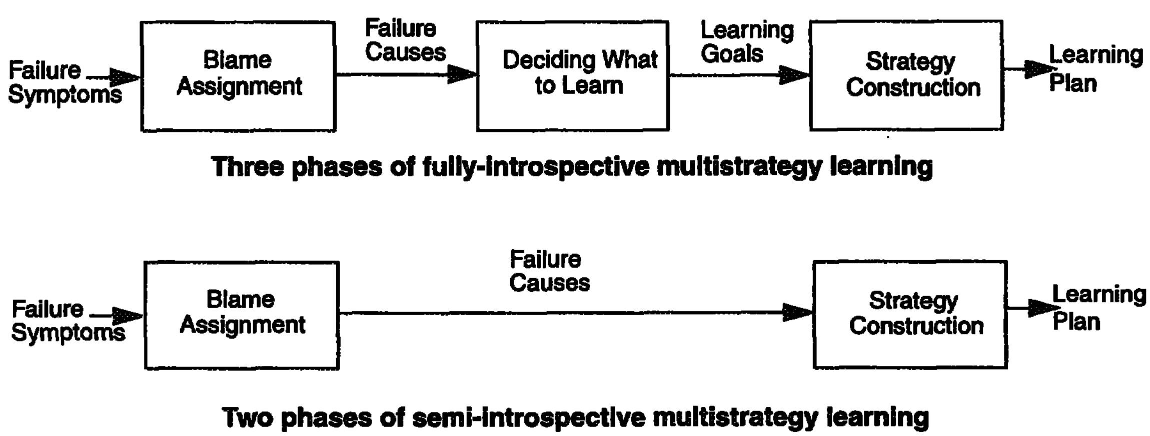

Figure 4. Learning goal ablation

approach (i.e., direct mapping from fault, or failure cause, to repair).

The final condition is called the no learning (NL) condition in which Meta-AQUA performs story understanding, but if a failure exists, the system constructs no learning strategy. This condition represents the baseline performance from which both the LG and RL conditions can be compared. Holding all parameters constant except the independent variables, Meta-AQUA is given input from the Tale-Spin problem generator and the dependent variable is measured.

The dependent variable must measure system performance (i.e., story understanding and explanation). In previous research, paraphrase and question answering tasks have been used as this measure (e.g., Lehnert, Dyer, Johnson, Yang, & Harley 1983; Schank & Riesbeck 1981; Wilensky 1978). If a reader sufficiently understands a body of text, the reader should be able to summarize the central points of the story and list the major events within it. If the story is well understood, then the reader can answer questions concerning the events and relationships within the story.

With story understanding programs such as BORIS (Lehnert et al. 1983), the researchers pose questions to the system and subjectively evaluate the answers to determine text comprehension effectiveness. One can count the number of questions answered correctly to ascertain an "absolute" measure of performance, but this is misleading. In contrast to externally posed questions, Chi (1995, Chi et al. 1989) reports that improved learning is correlated with human subjects who generate their own questions and explain the answers themselves. Being able to recognize that a gap exists in one's own knowledge, and thus to ask the question "Why don't I understand this?" (Ram 1991), is the first step to improving understanding. To pose self-generated questions thus indexes understanding and simultaneously reduces the probability of asking only easy questions. So, to evaluate the ability of the Meta-AQUA system, credit is given for simply posing a question that deserves asking.

Moreover, humans who are asked questions on reading tests are sometimes given points for partial answers. Unlike questions that have provably correct answers, answers to explanatory questions are difficult to judge in an absolute sense. So to be more realistic, the evaluation criterion in Meta-AQUA assigns credit for providing any answer to a question. Therefore, the full evaluation metric is as follows. For each anomalous or interesting input in a story, a point is given for posing a question, an additional point is given for providing any answer whatsoever, and a third point is assigned for answering what the researcher judges correct. The sum represents the dependent variable.

The Empirical Data

To serve as experimental trials and to minimize order effects, Tale-Spin generated six random sequences of Elvis-World stories. On each of these runs, Meta-AQUA processes a sequence three times, once for each experimental manipulation. The system begins all runs with the same initial conditions. For a given experimental condition, it processes all of the stories in the sequence while maintaining the learned knowledge between stories. At the end of the sequence, the system resets the BK. The input size for a run varies in length, but averages 27.67 stories per run.[7] The corpus for the six runs includes 166 stories, comprising a total of 4,884 sentences. The stories vary in size depending on the actions of the story and Tale-Spin's randomness parameters (e.g., the probability that a character will stop throwing an object on the current toss), but average 29.42 sentences.

Run Number Four. Run number four is particularly interesting because the greatest number of learning interactions occur in this set. The input to run four consists of 24 stories as enumerated in Table 1. The stories contain a total of 715 sentences, and the average number of sentences per story is 29.8. Each numeric entry in Table 1 contains a triple of the form <LG, RL, NL>. For example, the sixth column represents the number of learning episodes for each trial and for each condition. Note that the third element of each triple in this column is zero since learning is disabled in the NL condition. The fifth column (Question Points) contains the values for the dependent variable. These values represent the sums of triples from the second, third and fourth columns (Posed Questions, Answered Questions and Correct Answers, respectively).

In this run, random drug busts occur 11 times (5 with the canine squad and 6 with a lone police officer). Also, Dad is the most common protagonist, while Elvis, Officer1, and Lynn are tied for the least common. Furthermore, boredom is the major problem encountered by the main characters, although considering the number of random drug busts, the household can hardly be classified as sedate. The main characters solve (or attempted to solve) seven of these boredom problems by playing with one of three balls and solve three by playing with balloons. The state of being concerned is the least recurrent problem exhibited in the run.

[7] The reason that each run varies in length is that, after generating around 600,000 gensyms, Meta-AQUA will use all available swap space on the Symbolics and thus inadvertently halt the underlying LISP system. We then discard the story which is being processed at the time of the crash. The data from the remaining stories constitute the results of the run.

Table 1: Results from run number four

Story Number (sentences)[a]	Questions Posed (LG RL NL)	Answered Questions (LG RL NL)	Correct Answers (LG RL NL)	Question Points (LG RL NL)	Learning Episodes (LG RL NL)	Protagonist and Problem[b]
1 (26)	1 1 1	0 0 0	0 0 0	1 1 1	1 1 0	Mom bored (balloon)
2 (19)	3 3 3	3 2 2	1 0 0	7 5 5	2 3 0	Mom bored (ball)
3 (38B)	1 1 1	0 0 0	0 0 0	1 1 1	1 1 0	Elvis jonesing
4 (51b)	1 1 1	1 0 0	1 0 0	3 1 1	0 1 0	Dad jonesing
5 (21)	1 1 1	1 0 0	1 0 0	3 1 1	0 1 0	Mom bored (ball)
6 (13)	1 1 1	1 0 0	1 0 0	3 1 1	0 1 0	Officer1 concerned
7 (13)	1 1 1	1 0 0	1 0 0	3 1 1	0 1 0	Dad bored (ball)
8 (21)	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	Dad thirsty
9 (44B)	2 2 2	2 1 1	1 0 0	5 3 3	1 2 0	Dad thirsty
10 (51B)	3 3 3	2 1 1	2 1 0	7 5 4	0 1 0	Dad bored (balloon)
11 (11)	2 2 1	1 1 1	1 0 0	4 3 2	1 2 0	Lynn bored (ball)
12 (3)	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	Officer1 concerned
13 (47b)	2 2 1	1 1 0	1 1 0	4 4 1	0 0 0	Mom thirsty
14 (15)	4 4 4	4 2 3	4 0 0	12 6 7	0 4 0	Mom bored (ball)
15 (28)	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	Lynn jonesing
16 (42B)	2 2 2	2 1 1	2 1 0	6 4 3	0 1 0	Dad jonesing
17 (45b)	2 2 1	1 1 0	1 1 0	4 4 1	0 0 0	Elvis jonesing
18 (21)	2 2 2	2 1 1	2 1 0	6 4 3	0 1 0	Officer1 concerned
19 (20)	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	Dad jonesing
20 (52b)	2 2 1	1 0 0	1 0 0	4 2 1	0 1 0	Dad bored (balloon)
21 (39b)	2 2 1	1 1 1	1 1 1	4 4 3	1 1 0	Lynn jonesing
22 (17)	2 2 2	2 1 1	2 0 0	6 3 3	0 2 0	Dad bored (ball)
23 (40B)	2 2 2	1 1 1	1 1 0	4 4 3	1 1 0	Elvis thirsty
24 (38b)	2 2 1	1 1 0	1 0 0	4 3 1	0 1 0	Mom bored (ball)
Total 715	38 38 32	28 15 13	25 7 1	91 60 46	8 26 0	

a. The letter "B" means that the story contains an attempted drug bust by the police canine squad, whereas the letter "b" means that the officer entered the house alone to attempt a bust.

b. Items in parentheses represent games played to dispel boredom.

Table 2: Summary of results from run four

Learning Condition	Questions Posed	Answered Questions	Correct Answers	Total Question Points	Learning Episodes
LG	38	28	25	91	8
RL	38	15	7	60	26
NL	32	13	1	46	0

Table 2 summarizes the totals from Table 1. The dependent variable (column 5) shows that Meta-AQUA's performance under the LG condition is significantly greater than the performance under the RL condition. In turn, Meta-AQUA performance in the RL condition far exceeded the performance under the NL condition.

Alternatively, if only absolute performance (column 4) is considered, the differential is even greater. By this measure, the LG condition is more than three times the value of the RL condition, whereas, the performance of the NL condition is insignificant. By looking at column three, however, the numbers of questions answered in some way (right or wrong), are roughly equivalent in the RL and NL conditions, whereas the ratio of the LG condition to either of the other two is 2:1. Finally, the number of questions posed are virtually equal across all three conditions.

In contrast to these differences, Meta-AQUA attempts to learn from failure more than three times as often under the RL condition as under the LG condition. That is, learning is more *effective* with learning goals than without. In the RL condition, learning does not increase performance as much as does the LG condition, while concurrently, it leads Meta-AQUA to expend more resources by increasing the amount of learning episodes. Thus, the system works harder and gains less under RL than under LG.

Figure 5 graphs the accumulation of question points across trials (i.e., stories).[8] The behavior of the system as measured by the dependent variable is greatest under the LG condition, next best under RL, and worst under the NL condition. But, the trend does not hold for each trial. Figure 6 shows raw scores indicating that the NL condition actually outperforms the RL condition on trial number 14. The reason for this effect is that under worse-case conditions, if the interactions present between learning methods are negative, the performance may actually degrade. As a result, randomly ordered learning may be worse than no learning at all.

The differences as a function of the independent variable are even more pronounced if only accuracy (the number of correct answers) is examined and partial credit ignored. Figure 7 shows that under the RL condition, Meta-AQUA did not answer a question correctly until trial number 10, whereas under the NL condition, it did not perform correctly until trial 21. On the other hand, because under the LG condition the system learned a new explanation early in trial number 1, it was able to answer a question by trial number two. This striking result was facilitated by the random order of input (i.e., the second trial happened to be about the same problem as the first) as well as by computational introspection.

[8]Note that the final extent of all three curves reach the value of the triple in the totals column for column five.

Overall Results. Table 3 summarizes the evaluation data from the six program runs. As is evident across all runs, the LG condition consistently outperforms the RL condition in the total cumulative question points. In turn, the RL condition outperforms the NL condition, despite the occasional poor performance due to negative interactions. As indicated by the standard deviations, the amount of differences between and within conditions exhibit high variability across runs.

Given these totals, the percent improvement for either learning condition over the NL base condition is simply the ratio of the difference in the base performance score and either score to the base score itself. Thus for run one, the ratio of the difference between the LG and NL conditions (35 points) to the NL condition (50 points) is .7, or 70 percent. Again, the improvement in performance for the LG condition is consistently higher than that of the RL condition. This difference is calculated in the final column. The differential is the percent improvement of the LG condition over the RL condition and is computed by the same measure as was the improvements in the individual learning conditions. That is, the differential is the ratio of the difference between the two improvements to the lower rate.[9] Thus, the differential between the LG rate of learning in run number one and that of the RL condition is the ratio of the difference (8 percentage points) to the RL percentage (62). Hence, the ratio is .129, or an improvement of nearly 13 percent.

Although the average differential between the two learning conditions (i.e., between fully-introspective and semi-introspective multistrategy learning) is more than 106 percent with a large standard deviation, this figure still overstates the difference. The expected gain in learning is more conservative. The differential between the average LG improvement (102.70) and the average RL improvement (65.67) is a 56.38 percent difference. That is, across a number of input conditions, the use of learning goals to order and combine learning choices should show about 1.5 times the improvement in performance than will a straight mapping of faults to repairs when interactions are present.

[9]Note that this ratio can also be calculated as the difference between the performance scores of the learning conditions to the difference between the performance score of the RL and NL conditions. In other words, the ratio (LG-RL) / (RL-NL).

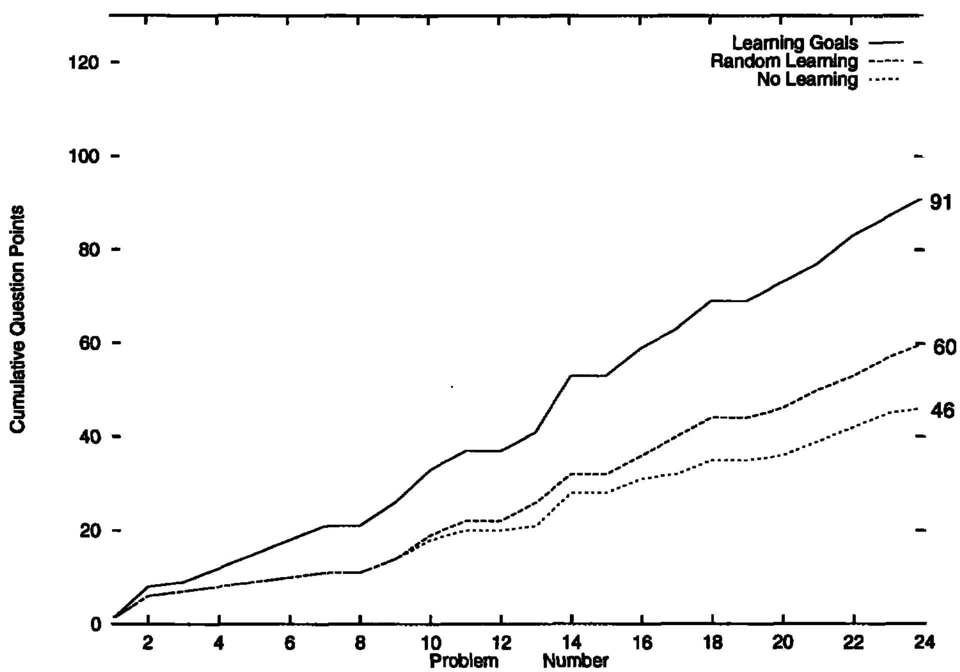

Figure 5. Run 4, cumulative question points as a function of the number of problems

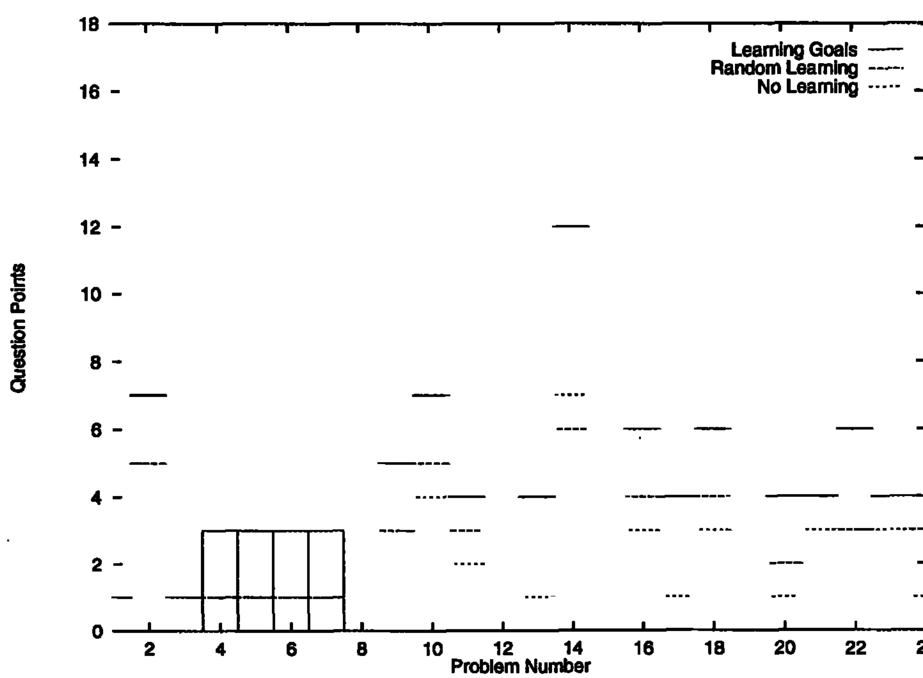

Figure 6. Run 4, question points histogram

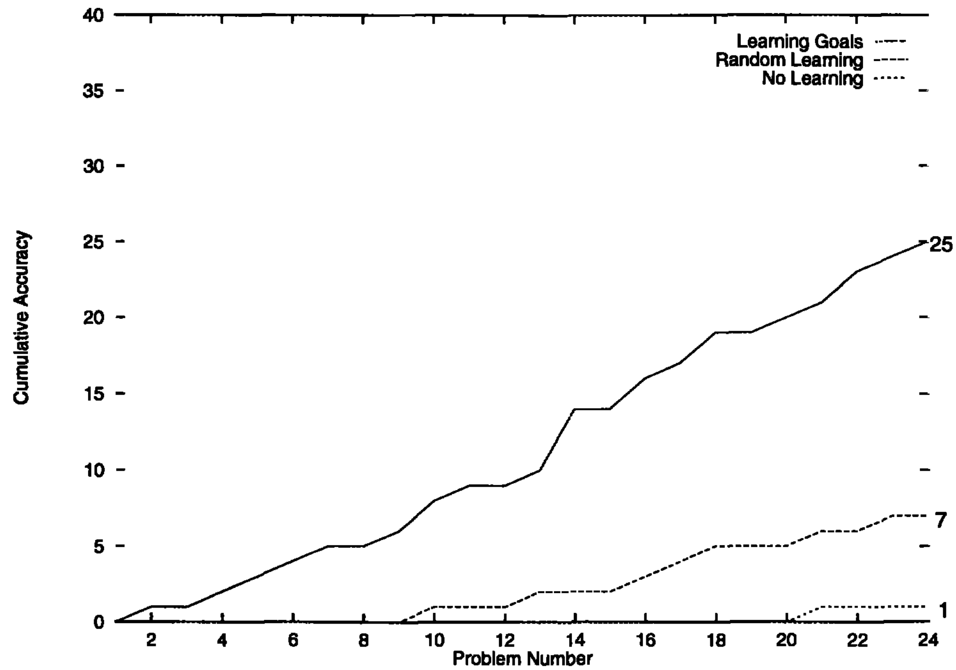

Figure 7. Run 4, cumulative correct answers (accuracy) as a function of the number of problems

Table 3: Summary of cumulative results

Run Number[a]	Cumulative Question Points LG	RL	NL	% LG Improved	% RL Improved	Improvement Differential %
Run 1 (34)	85	81	50	70.00	62.00	12.90
Run 2 (30)	106	98	43	146.51	127.91	14.55
Run 3 (28)	120	102	60	100.00	70.00	42.86
Run 4 (24)	91	60	46	97.83	30.43	221.43
Run 5 (22)	57	49	27	111.11	81.48	36.36
Run 6 (28)	103	66	54	90.74	22.22	308.33
Averages	93.67	76.00	46.67	102.70	65.67	106.07
Std. Dev.	21.72	21.31	11.34	25.43	38.17	126.59

a. Amounts in parentheses indicate total number of stories in each run.

Summary and Discussion

The experiments reported in this paper provide a number of results that support the hypothesis that computational introspection facilitates multistrategy learning. Meta-AQUA expended more learning resources and induced less performance improvement without learning goals than it did under a condition that included them. Moreover, we have shown that because learning algorithms negatively interact, the arbitrary ordering of learning methods (i.e., as under the RL condition) can actually lead to worse system performance than no learning at all. Therefore, an explicit phase to decide exactly what to learn (i.e., to spawn learning goals or an equivalent mechanism) is *necessary* to avoid these interactions and to maintain effective learning in multistrategy environments. The paper also provided a novel quantitative measure with which to evaluate the comprehension process. As dependent variable, this partial credit metric provides rewards for both posing questions autonomously and giving some type of answer, as well as for getting answers correct.

Because of the considerable computational overhead involved in maintaining a reasoning trace, performing blame-assignment, spawning learning goals, and constructing a plan with which to pursue such goals, the benefits of using introspection must be substantial to justify the costs.[10] Furthermore, under extremely complex situations or in informationally impoverished circumstances, deciding on an optimal learning goal is certainly intractable. In such situations, it may be more beneficial to proceed without further reasoning, rather than to attempt to understand the exact causes of the failure. Knowing when a learning task is worth pursuing is itself an important skill to master for an intelligent system. Identifying the most appropriate conditions for the use of an introspective approach is therefore a desirable research goal. To establish only that introspection facilitates learning and that the model of introspection has some quality of reasonableness is not satisfactory. Although further inquiry into these conditions is left for future research, a number of remarks can be made at this time.

If the distributions of the kinds of failures generated by the performance task change the nature of the differences in the learning curves generated in the experiments used to establish the hypothesis, then applicability conditions can be established that predict when the utility of introspection exceeds its cost. The space of applicability conditions for introspection is expected to emerge from the taxonomy of causal factors presented in Ram, Cox, and Narayanan (1995). It has already been shown through the existing implementation that introspection in certain circumstances is tractable. Thus, a lower bound is already available. It is clearly not possible to reason in any effective manner if all possible failures occur at once or given an overly-sparse BK. So an analysis of the interaction of the taxonomized causal factors should result in a set of complex failures that can be programmed into Tale-Spin in order to produce various distributions of errors. Meta-AQUA is expected to have difficulty learning from some of the failure combinations within these error distributions. As with the ablation study, measures with and without introspection provide the independent variable for the evaluation of learning. The results should itemize the conjunctions of failure from which it is impossible to recover and those for which a reflexive or tightly coupled approach is more suited.

In the interim, a potential heuristic for deciding when to use an introspective approach is to qualitatively ascertain whether or not interactions between learning mechanisms available to the learner exist. If they exist, then the approach should be applied, otherwise a more reflexive approach is licensed. In speculation, another potential heuristic for determining that introspection is a win is to use a threshold for the number of failure symptoms above which introspection will not be attempted. Through experimentation, this threshold number should be obtained empirically given a distribution of known problem types and a random selection of problems from the distribution. The identification of such heuristics will enable the practical use of introspective methods in systems that cannot afford to squander precious resources with intractable computation.

Acknowledgments

This research is supported by AFOSR under contract #F49620-94-1-0092 and by the Georgia Institute of Technology. I thank Ashwin Ram, Janet Kolodner, Mimi Recker, Kurt Eiselt, Tony Simon, Kenny Moorman, Juan Carlos Santamaria, Tucker Balch, and the anonymous reviewers for comments on various drafts of this paper. I also wish to gratefully acknowledge Mark Devaney's assistance in reprogramming Tale-Spin.

---

[10]One must be cautious, however, when summarily dismissing introspection due to computational overhead costs alone. Doyle (1980) warns that to disregard the introspective component and self-knowledge in order to save the computational overhead in space, time, and notation is discarding the very information necessary to avoid combinatorial explosions in search (p. 30).

References

Carbonell, J. G. (1986). Derivational Analogy: A theory of Reconstructive Problem Solving and Expertise Acquisition. In R. Michalski, J. Carbonell and T. Mitchell, eds., *Machine learning II: An Artificial Intelligence Approach*, 371-392. San Mateo, CA: Morgan Kaufmann Publishers.

Chi, M. T. H. 1995. Revising the Mental Model as One Learns. Plenary address to the Seventeenth Annual Conference of the Cognitive Science Society. Pittsburgh (July 23).

Chi, M. T. H., Bassok, M., Lewis, M., Reimann, P., and Glasser, R. 1989. Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. *Cognitive Science* 13:145-182.

Cox, M. T. 1994. Machines that Forget: Learning from Retrieval Failure of Mis-Indexed Explanations. In A. Ram and K. Eiselt, eds., Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society, 225-230. Hillsdale, NJ: Lawrence Erlbaum Associates.

Cox, M. T. 1996. Introspective Multistrategy Learning: Constructing a Learning Strategy under Reasoning Failure. Ph.D. diss., Technical Report, GIT-CC-96-06, College of Computing, Georgia Institute of Technology, Atlanta. Available URL: ftp://ftp.cc.gatech.edu/pub/ai/ram/git-cc-96-06.html

Cox, M. T., and Ram, A. 1995. Interacting learning-goals: Treating learning as a planning task. In J.-P. Haton, M. Keane and M. Manago, eds., *Advances in case-based reasoning*, 60-74. Berlin: Springer-Verlag.

Doyle, J. 1980. A Model for Deliberation, Action, and Introspection. Ph.D. diss., Technical Report, TR-58, Department of Computer Science, Massachusetts Institute of Technology, Cambridge.

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

Lehnert, W., Dyer, M. G., Johnson, P., Yang, C., and Harley, S. 1983. BORIS - An Experiment in In-Depth Understanding of Narratives. *Artificial Intelligence* 20(1): 15-62.

McCarthy, J. 1959. Programs with Common Sense. In Symposium Proceedings on Mechanisation of Thought Processes, vol. 1, 77-84. London: Her Majesty's Stationary Office.

Meehan, J. 1981. Talespin. In R. C. Schank and C. Riesbeck, eds., *Inside Computer Understanding: Five Programs Plus Miniatures*, 197-258. Hillsdale, NJ: Lawrence Erlbaum Associates.

Michalski, R. S. 1994. Inferential theory of learning: Developing Foundations for Multistrategy Learning. In R. Michalski and G. Tecuci, eds., *Machine learning IV: A Multistrategy Approach*, 3-61. San Francisco: Morgan Kaufmann.

Minsky, M. L. 1965. Matter, Mind, and Models. In Proceedings of the International Federation of Information Processing Congress 1965, vol. 1, 45-49. (Original work from 1954)

Pazzani, M. 1994. Learning Causal Patterns: Making a Transition from Data-Driven to Theory-Driven Learning. In R. Michalski and G. Tecuci, eds., *Machine learning IV: A Multistrategy Approach*, 267-293. San Francisco: Morgan Kaufmann.

Ram, A. 1991. A Theory of Questions and Question Asking. *The Journal of the Learning Sciences* 1(3&4): 273-318.

Ram, A., and Cox, M. T. 1994. Introspective Reasoning Using Meta-Explanations for Multistrategy Learning. In R. S. Michalski and G. Tecuci, eds., *Machine learning IV: A Multistrategy Approach*, 349-377. San Francisco: Morgan Kaufmann.

Ram, A., Cox, M. T., and Narayanan, S. 1995. Goal-Driven Learning in Multistrategy Reasoning and Learning Systems. In A. Ram and D. Leake, eds., *Goal-Driven Learning*, 421-437. Cambridge, MA: MIT Press/Bradford Books.

Schank, R. C., and Riesbeck, C. eds. 1981. *Inside Computer Understanding: Five Programs Plus Miniatures*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Veloso, M. (1994). *Planning and Learning by Analogical Reasoning*. Berlin: Springer-Verlag.

Wilensky, R. 1978. Understanding Goal-Based Stories. Ph.D. diss., Technical Report, 140, Department of Computer Science, Yale University, New Haven, CT.