

# **Introspective Multistrategy Learning**

Michael T. Cox  
March 16, 1993



**Cognitive Science**  
**Report No. 2**

**GEORGIA INSTITUTE  
OF TECHNOLOGY**

A UNIT OF THE UNIVERSITY SYSTEM OF GEORGIA

# Introspective Multistrategy Learning

**Michael T. Cox**

Artificial Intelligence / Cognitive Science Group  
College of Computing  
Georgia Institute of Technology  
Atlanta, GA 30332-0280  
email: cox@cc.gatech.edu

*Ph.D. Thesis Proposal*

December 2, 1992

*Final Revision:*

March 16, 1993

*Proposal Committee:*

Dr. Ashwin Ram\* (College of Computing)

Dr. Janet Kolodner (College of Computing)

Dr. Tony Simon (College of Sciences, School of Psychology)

---

\*. Thesis Advisor.

## **Abstract**

The thesis of this proposal is that introspection facilitates learning by providing a basis for identifying what needs to be learned and for selecting an appropriate learning algorithm. If the system has a model of its own reasoning processes and of the knowledge used by these reasoning processes, it can declaratively represent the events and causal relations in the mental world in the same manner that it represents events and relations in the physical world. A multistrategy learning system, in which several learning algorithms are available, can decide what to learn, and which algorithm(s) to apply, by analyzing this model of its reasoning. This introspective analysis allows it to understand its reasoning failures, to determine the causes of the failures, to identify needed knowledge repairs in order to avoid such failures in the future, and to select the learning algorithm appropriate for the needed repairs. Thus, the object of the proposed research is to develop both a content theory and a process theory of introspective multistrategy learning and to establish the conditions under which such an approach is fruitful.

# Table of Contents

Abstract . . . . .	i
Table of Contents . . . . .	ii
List of Figures . . . . .	iv
List of Tables . . . . .	iv
1 INTRODUCTION . . . . .	1
2 PROCESS AND CONTENT THEORIES . . . . .	6
3 REASONING MODEL . . . . .	11
3.1 Theoretical Assumptions . . . . .	11
3.2 Process Theory of Understanding . . . . .	14
3.3 Content Theory of Understanding . . . . .	21
3.4 Taxonomy of Reasoning Failures . . . . .	26
4 LEARNING MODEL . . . . .	30
4.1 Implementation and Example . . . . .	33
4.2 Process Model of Learning . . . . .	36
4.2.1 Operationalized Phases of Learning . . . . .	40
4.2.2 Process Theory Functional Arguments . . . . .	41
4.3 Content Theory of Introspective Explanations . . . . .	43
4.3.1 Base Class IMXPs . . . . .	45
Successful Prediction . . . . .	46
Inferential Expectation Failure . . . . .	47
Retrieval Failure . . . . .	48
Incorporation Failure . . . . .	49
4.3.2 Core Class IMXPs . . . . .	50
Erroneous Association . . . . .	50
Missing Association . . . . .	50
Novel Situation . . . . .	53
Incorrect Domain Knowledge . . . . .	53
4.3.3 Composite Class IMXPs . . . . .	54
5 CONCLUSIONS . . . . .	58
5.1 Comparison of Learning and Understanding . . . . .	58
5.2 Artificial Intelligence Related Research . . . . .	62
5.3 Psychological Influences and Support . . . . .	63

A APPENDIX: Research Agenda . . . . .	.68
A.1 Evaluation . . . . .	.68
A.1.1 Computational Empirical Evaluation . . . . .	.69
A.1.2 Cost-Benefit Analysis . . . . .	.70
A.1.3 Psychological Empirical Evaluation . . . . .	.71
A.2 Plan . . . . .	.72
ACKNOWLEDGEMENTS . . . . .	.74
REFERENCES . . . . .	.75
Index . . . . .	.82

## List of Figures

Figure 1: The Case-Based Generation Process .....	9
Figure 2: Assumptions .....	11
Figure 3: Augmented Generate-and-Test Cycle .....	15
Figure 4: Question-Driven Understanding .....	18
Figure 5: Phases of Understanding .....	23
Figure 6: Decide-Compute-Node .....	25
Figure 7: The Drug Bust Story .....	33
Figure 8: Phases of Learning .....	37
Figure 9: Introspective Multistrategy Learning Algorithm .....	39
Figure 10: Successful Prediction .....	47
Figure 11: Instantiated Successful Prediction .....	47
Figure 12: Expectation Failure .....	48
Figure 13: Instantiated Expectation Failure .....	48
Figure 14: Retrieval Failure .....	49
Figure 15: Instantiated Retrieval Failure .....	49
Figure 16: Incorporation Failure .....	50
Figure 17: Missing Association Core Type .....	52
Figure 18: Instantiated Composite Type .....	55
Figure 19: The Parallels in Learning and Understanding .....	61

## List of Tables

Table 1: Dimensions of Failure .....	27
--------------------------------------	----

# 1 INTRODUCTION

Simply stated and in the narrowest sense, the central problem addressed by this proposal is that of *strategy selection* in machine learning, particularly, in failure-driven learning. That is, given some computational task (e.g., story understanding or problem solving) specified by the system's goals, context and some input, if a failure occurs during the task, the problem is to choose a learning algorithm with which to repair the background knowledge<sup>1</sup> (BK) of the system. The BK is considered repaired if, given a similar future situation, the failure will not recur.<sup>2</sup> In the broadest interpretation, this research presents a theory of self-understanding and its relation to human learning. Although many of the issues of machine learning and human learning are separate and incommensurable, this thesis will make bridges where possible, taking constraints from known human limitations and abilities and attempting to form a computational model of learning that is sufficient for any intelligent agent.

Recent attention to multistrategy learning systems is evident from numerous sources in the machine learning literature (e.g., Carbonell, Knoblock & Minton, 1991; Michalski & Teccuci, 1991, to appear) and some in the psychological literature (e.g., Anderson, 1983; Wisniewski & Medin, 1991). Multistrategy learning systems are those that integrate various learning algorithms into a unified whole, and thus contrast with single-strategy systems such as Soar (Laird, Rosenbloom & Newell, 1986) in which all learning is performed by a single learning mechanism. Whereas all learning in Soar reduces to chunking, methods as disparate as explanation-based learn-

1. The background knowledge includes more than simply domain knowledge. It can also contain knowledge such as metaknowledge, heuristic knowledge, associative knowledge, and knowledge of process.
2. The Inferential Learning Theory of Michalski (1991) has defined a learning task as consisting of three components: some input (information), a BK, and a learning goal. The learning goal determines the relevant pieces of the input, the knowledge to be acquired, and the criteria for evaluating the learning. The model of learning presented here is consistent with these constraints.

ing, similarity-based learning, deduction, abduction, constructive induction, and analogy can be included in the same multistrategy framework. Some multistrategy systems combine multiple algorithms in a cascade or piped-flow arrangement. The output of algorithm  $n$  becomes the input to algorithm  $n+1$ . Other systems use a combination of algorithms on a given input in some predetermined fashion. Each algorithm calculates a part of the overall solution. These systems are usually fixed in their means of processing. A third type of multistrategy learning explicitly selects and invokes various learning algorithms as a function of the characterization of the input. The system presented here is in the latter category.

Research into multistrategy learning is necessary on pragmatic grounds when complex worlds are the domains of learning systems. Such approaches allow for maximal flexibility. Significant interactions are present in multistrategy systems, however, that are not apparent in isolated systems. For example, if two algorithms modify the domain knowledge of the system, and a dependency exists between the two, such that one strategy modifies a part of the domain knowledge that the second one uses, then an implied sequencing must be enforced: the first strategy must be applied before the second. Such dependencies do not exist in single-strategy systems. Research into multistrategy systems contributes to the resolution of many such complexities in real-world systems.

The theory presented in this work is interesting because the choice of algorithm is not simply a function of the input, where the input is some set of assertions about the world, or even a faulty solution tree, but rather the input to the learner represents a trace or declarative representation of the reasoning itself that produced the solution. The choice of a learning strategy is therefore a function of the prior reasoning that produced the error, as much as it is a function of the erroneous solution.<sup>3</sup> The solution to a problem is usually a structured set of operations that institutes changes in the world, such as chess moves that modify an external board position; whereas a reasoning trace is a structured set of mental operations that produces changes in states of the mind, selects problem

operators, and eventually results in the solution plan. Thus to decide on a choice of strategies, our theory of learning depends on introspection of the mental world, as much as it depends on an analysis of both the problem and the solution in the external world. In contrast, systems that make decisions based on a solution alone have only an indirect relationship to the actual causes of the failure.

Yet it is not immediately apparent why introspection is necessary, or even desirable, at least not with respect to solving the strategy selection problem. Introspection has a number of disadvantages including considerable computational overhead. In the realm of general reasoning, though, adding introspection to a machine allows it to have an idea of what it is doing and why. A machine applying deductive theorem proving certainly does not understand theorem proving in the manner that a mathematician does. No model of the problem solving process itself exists. The machine does not understand theorem proving, though it can perform theorem proving. Moreover, with respect to learning, there are additional arguments in favor of introspection.<sup>4</sup> To justify the importance of introspection in multistrategy learning systems, this proposal advances the following argument: To choose an algorithm, a system needs to know what is supposed to be learned; to decide what needs to be learned, it must know the cause of failure; to determine the cause of the failure, it must perform blame assignment; and to perform complete blame assignment in many situations, it must reflect upon its own reasoning.

To properly select an algorithm or learning strategy, the system must know what it needs to learn; it must have a *learning goal* or target. Imagine that the learning strategies from which the system

---

3. Carbonell (1986) argues that a key insight into analogical reasoning is that solution derivations contain useful information beyond the information in the solution itself. His method is to map the derivation of old solutions onto new problems, rather than map old solutions into new solutions. This insight was one of the first arguments in favor of maintaining reasoning traces.

4. Of course it is a well-founded fact that the veracity of human introspection is limited. We are not claiming that introspection is a computational panacea, rather this research investigates the role of introspection in learning and the conditions under which it is either a gain or a loss. See section A, "APPENDIX: Research Agenda," starting on page 68.

chooses are like operators in planning paradigms (Hunter, 1990; Ram & Hunter, 1992). To select an operator effectively in planning systems, the system must have a goal toward which operators make progress; thus, selecting the actions that constitute steps of a plan is based on the goal of the system. Since operators have results and preconditions, they can be chained such that various operators are chosen on the basis of resultant states that satisfy the preconditions of, and therefore enable, other operators. Thus, they can be chained to produce a series of plan steps that eventually matches the plan goal. Similarly, as plan steps produce changes in the world, learning strategies produce changes in the system's BK. To produce productive changes in the BK, then, the system must have an appropriate learning goal.

Furthermore, to generate the learning goal, the system must know the cause of the failure; it must perform *blame assignment*. Blame assignment (or, conversely, credit assignment) is a well-known problem, going back as far as Minsky (1963), involving the construction of explanations for how and why a failure occurs (or how and why success occurs). Without having knowledge of what caused the system to fail at its reasoning task, it is difficult to know what to learn to avoid subsequent failures in like situations. Perhaps bottom-up reinforcement schedules can help the system learn what to do without it knowing why, but surely no deliberative methods will be able to form a goal to modify the BK in any meaningful way without first analyzing the failure. Explanation is therefore crucial in fully understanding the relation between the current state of the system, its BK, and the current condition of the external world. In this way blame assignment can be viewed as a special form of abduction.

To perform effective blame assignment, the system must be able to reason about its own reasoning, in addition to reasoning about the world or the results of its own reasoning. Determining the reasons why failure occurs is often not simply a matter of understanding events in the world, or even the plans created, rather failures can be attributable to the reasoning process, or the choice of one.

Newell and Simon (1972) show that human subjects often make reasoning mistakes because of the wrong choice of reasoning strategy. Given the "Magic Squares" word problem, such that the numbers in some matrix must add up across, up, down, and diagonally, the solution is quite easy using an analogy to tic-tac-toe, but is extremely difficult using means-ends-analysis. Unfortunately, most subjects use this latter reasoning and therefore cannot solve the problem. To explain this problem effectively, it is useful to have a mental interpretation of the problem solving process, as well as an explanation that deals with the problem itself.

As another case of the intertwined relationship between blame assignment and introspection, consider the stranded motorist example (Cox & Ram, 1992b). If an agent runs out of gas on a vacation, a number of causes could have contributed to the failure. A problem could have occurred with the car's fuel system (perhaps a hole developed in the gas tank or fuel lines), or a problem could have occurred with the driver's memory system (perhaps the agent forgot to fill up with gasoline before starting his trip). If the agent is aware of his prior reasoning, including the formulation of a goal to fill up the tank, then when the car rolls to a stop, he should be reminded of the suspended planning goal. The blame is thus associated principally with the mental faculties and the indexes that address the forgotten task, rather than with the physical operation of the car, although there is an unmistakable interaction between the two.<sup>5</sup> One important type of introspection is to realize that the cause of failure was not the plan or solution generated by the reasoner before the trip, but instead was the memory system and the organization of the knowledge that together did not retrieve the suspended goal, given the state of being at or near the gas station. Rather than improve the plan itself, such an analysis can allow a system to improve the organization of the BK by learning better indexes for particular types of suspended goals.

---

5. Without some naive knowledge of the physical model of the car, the act of filling the gas tank is meaningless, thus memory for performing it is mechanical at best.

Therefore, in many situations a model of introspection is required to perform blame assignment. Blame assignment is crucial in choosing a learning goal, and the choice of a learning algorithm depends on the learning goal. The solution to be developed in the following sections is to represent the reasoning in a declarative trace, then to reason about the trace. To do this one must have a model of reasoning and a vocabulary with which to express inspectable instances of reasoning. The system can then perform blame assignment and ultimately choose the proper learning algorithm with which to repair the BK of the system. Section 2 discusses issues in both process theories and content theories of cognition that impinge upon the construction of such models. Section 3 develops an explicit model of reasoning, giving both a process account that models intelligent understanding and a content theory of vocabulary. Section 4 describes the process and content theory of learning and its relation to introspection. Examples are presented in an implementation called Meta-AQUA. Preliminary conclusions and related research are presented in Section 5. Appendix A footnotes the proposal by providing criteria from which to judge the theory and developing a preliminary research agenda and a plan for bringing the thesis to closure.

## **2 PROCESS AND CONTENT THEORIES**

*Content theories* provide the vocabulary and structure for representing knowledge, whereas *process theories* specify the transformations performed on such knowledge (Birnbaum, 1986; Domeshek, 1992). Content theories provide a component theory that specifies the objects or components in the domain and the features that best describe the components. Also, a content theory provides constraints and inferential relationships between the features. Content theories therefore possess commitments to both domain ontology as well as domain physics (Domeshek, 1992). Because the domain of this research is reasoning itself, rather than some external behavior, our con-

content theory is unique in that it becomes a descriptive language of the processes found in our process theory. The intent of this paper is to outline a broad theory of introspection, understanding, and learning, by providing specific commitments as to the kind of processes that might account for such cognitive activities, and the kind of representational language that might be suited for computationally describing and making inferences from these phenomena. Using such a framework this thesis will prescribe a solution to the strategy selection problem.

A typical cognitive theory accounts for a specific class of tasks in a particular domain of human endeavor requiring reason. For example, given the task of design and the domain of meal planning, a content theory provides a language that adequately describes objects and events in the world of meal planning and design in general, while a process theory specifies the mental processes involved in design. The process theory then explains and predicts the behavior of agents performing meal planning activities, such as transforming meal specifications and constraints (described with the content theory) into artifacts of the domain (in this case, a menu of courses to be prepared). When preparing dinner for both a meat and potato lover and a vegetarian, the process theory enumerates the kinds of reasoning performed by the designer/planner, given the current contents of the refrigerator and the kitchen cupboards. The theory would dictate the transformations necessary to generate a meal plan and would specify the connectivity between cooperating processes. The objects in the content theory (domain knowledge of design and meals) and processes in the process theory (the transformations on such knowledge) are thus related, but quite distinct.

In a theory of introspection, however, the content and process theory are much more intimately related. The content theory must be able to represent the events that the process theory describes. This self-referential constraint is present because introspection, by definition, is thinking about thinking, and thinking about one's own knowledge and memory. Now, if the system is to process memories of its own processing, then a language is needed with which to represent the processing

itself. During reflection the processes transform and operate upon descriptions of themselves, so if the external domain of this thesis were to actually be meal design, one would need to specify the processes that account for design, as would a standard theory of design, but the content theory would concentrate on representations of these processes; a content theory of culinary objects and events would be secondary.

JULIA (Hinrichs, 1992), a case-based meal planner, implements the processes depicted in figure 1. A design goal is input into an analysis process that determines an appropriate search method. The search method is given to the retrieval process, which finds a relevant past case from the system's case memory. An adaptation process then generates the proper mappings and instantiates values to use in the new situation. An adapted meal plan is then returned as a result, or, if the plan is insufficient, the problem is reformulated and the process restarted.

JULIA's content theory of meal design provides the vocabulary used to describe cases of meal planning, such as the courses in a meal, meal ingredients, constraints on the design, goals of the planner, and preferences of the diners. To reason explicitly about the process of meal planning, however, a system must be able to represent not only the final result of a meal design, but it must also possess a way of representing the design event that produces the meal plan. It is not sufficient to simply annotate the final solutions (meal cases) with features signifying what occurred during the meal-planning process. The most obvious reason for precluding such an approach is that the process of meal planning is recursive; that is, a plan may be generated by an arbitrary number of passes through the planning loop. An annotated meal plan is therefore insufficient to distinguish between the various activities at similar points in the process.

Instead of simple annotations, it is desirable to create a chain of structures or nodes, one for each process in the planning effort. Each node would record the input and output, the bases and context

for its results, and a link to the following process. In this way the system could represent, for example, an initial analysis, a retrieval, and an adaptation, producing a case that did not work, then a reformulation of the problem followed by another series of analyze, retrieve and adapt steps. A benefit of producing this record is that it is also available for use by subsequent processes in the planning mechanism. JULIA's strategy, in the face of failure, is to loosen the constraints on a problem. A possibility exists that the problem specification was not at fault, however, but that the choice of search method was faulty instead. So rather than reformulating the problem after failure, another search method may be chosen to find a more appropriate case to adapt, given the original formulation of the problem. Thus, JULIA might be able to learn search strategies, as well as acquire new cases. Regardless, by recording the design process and representing it explicitly, far more information is available with which to plan as well as learn.

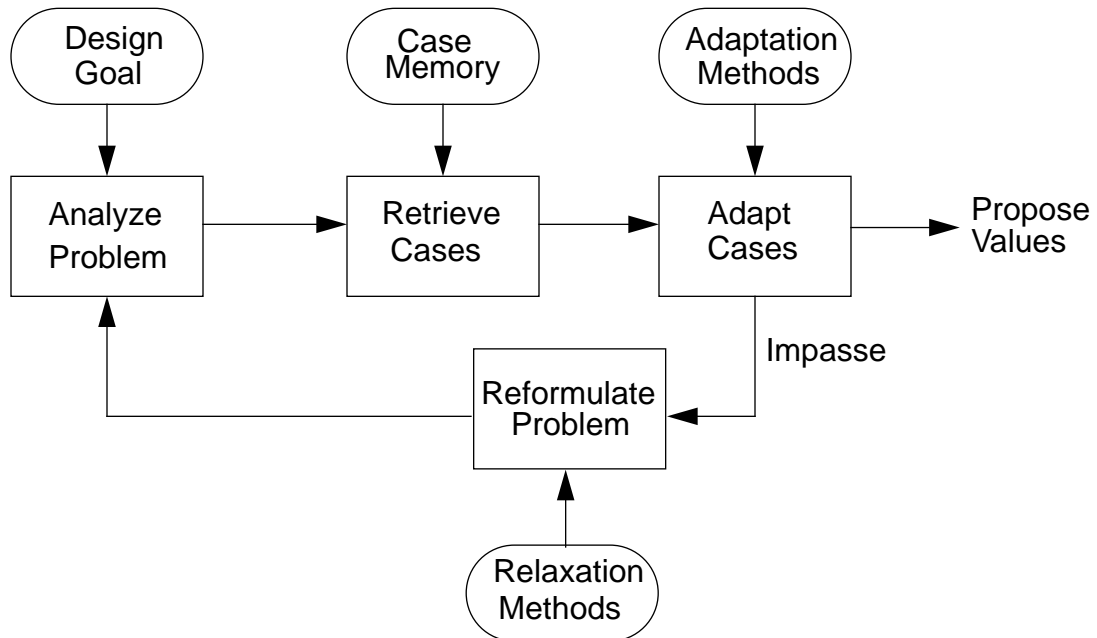


Figure 1: The Case-Based Generation Process (Adapted from Hinrichs, 1992)

The approach this paper will take, then, is consistent with the above analysis. We will develop a specific model of reasoning, along with a representational language and a knowledge taxonomy for expressing instances of reasoning. Once expressed in some declarative, inspectable form, the system can then process instances of its own reasoning in much the same manner as it processes input from the world.

The type of reasoning of primary concern is that of understanding and comprehension of a given input. As opposed to the JULIA example, however, the primary reasoning goal is to choose a reasoning method best suited for understanding the input. To understand the input then requires using the selected method to build an explanation of the input with regard to the knowledge of the system and the context in which the input appears. When these explanations fail, the learning task involves explaining faulty reasoning that contributes to failed explanations; thus, the system must explain its own explanations. A knowledge structure called a *Meta-Explanation Pattern (Meta-XP)*, which is an abstract causal pattern that represents how and why certain conclusions are drawn by the reasoner, is used to perform this task (Cox, 1991; Cox & Ram, 1991, 1992a; Ram & Cox, to appear). Meta-XP theory is a content theory that attempts to use meta-explanations to build the knowledge structures used for explaining most classes of reasoning failures and to use these constructions for choosing a learning algorithm. A system that possesses a library of such structures, and that can retrieve, adapt, and apply them to a given reasoning failure, is using a case-based approach to introspection and learning.

In open-world scenarios with many possible sources of failure, a number of machine learning techniques may apply to a given learning situation. Meta-AQUA is a learning system designed to test our theory of introspective multistrategy learning under such conditions. It contains various learning algorithms such as explanation-based generalization, similarity-based learning, and index learning. Treating the learning as a planning task, the system can post and order learning goals to

perform given learning tasks and choose an algorithm to accomplish the goals. In such a paradigm the management of such learning goals becomes a focus as in traditional planning systems. The following sections explore some issues, problems and solutions in such a framework.

### **3 REASONING MODEL**

This section states its suppositions and then draws both a process and content theory of reasoning in the form of the task of understanding that follows from the assumptions. It concludes with a taxonomy of reasoning failures based on the theory of understanding.

#### **3.1 Theoretical Assumptions**

The results, conclusions and the very structure of this theory depends on the broad assumptions enumerated in figure 2.

- Reasoning is goal-directed processing of input given some knowledge.
- Multistrategy reasoning is appropriate for both understanding and learning.
- Knowledge is memory-based.
- Learning is failure-driven.

Figure 2: Assumptions

First and foremost, we assume that cognition is essentially goal-directed processing of a given input using the reasoner's knowledge. Our focus is therefore on the deliberative and top-down

components of thought, rather than the data-driven or situation-specific factors. This thesis will not deny that bottom-up factors affect both reasoning and learning, though, as a research strategy, these factors will be minimized or ignored. Such a position is consistent with traditional cognitive science perspectives, although it is indeed at odds with some recent stances, such as the situated cognition paradigm (Suchman, 1987; Clancey, 1991).

We also assume, given the general arguments in the introduction, that the multistrategy framework is appropriate for cognitive tasks. The generic task view of Chandrasekaran (1989) provides additional support for this assumption by arguing that various general methods exist that apply to various problem solving tasks. More than one method may apply to a given task or subtask, so strategy selection is unavoidable in problem solving, whether performed by the knowledge engineer or the system itself.<sup>6</sup> Therefore, extending this idea further, rather than one basic mechanism, many different processes and algorithms can account for cognition in general and learning in particular. The strategy selection problem is thus pertinent to both reasoning and learning.

A third assumption is that the reasoner's knowledge is memory-based, and therefore subject to storage and retrieval constraints, particularly the indexing problem. The indexing problem (Domeshek, 1992; Kolodner, 1984; Schank, 1982; Schank and Osgood, 1990) is the problem of choosing cues, or features of an input, to be used as indexes for retrieving from memory the knowledge structures necessary to process an input. Thus, in such memories, knowledge organization is a large concern in both reasoning and learning functions.

Related to the above premise is the decision to bisect knowledge into two parts. The system's *background knowledge* (BK) contains representations for all long term knowledge, such as conceptual knowledge, episodes and cases, control knowledge or heuristics, beliefs, and knowledge about its

---

6. For specific implementations of this view see Goel & Callantine (1991) for descriptions of ROUTER and Punch (1991) for a discussion of the TIPS system.

own knowledge and reasoning processes (metaknowledge). In contrast to the BK, in our formulation another kind of knowledge exists, called the *foreground knowledge* (FK). The FK constitutes the current model of the input that has been constructed, and the memory of the reasoning with which such a model was built. Goals that are spawned during construction of this reasoning belong in the FK also. If the goals cannot be achieved immediately, however, they may be transferred to the BK, along with a trace of the reasoning that spawned them, in order to suspend the processing until their achievement is more likely (e.g., when the preconditions upon which they depend become available). In such opportunistic reasoning (Birnbaum & Collins, 1984; Hammond, 1988; Hayes-Roth & Hayes-Roth, 1979; Ram, 1989) the goals are indexed in the BK by features selected to match cues in the environment that are characteristic of conditions likely to exist when resumption of the goal processing is profitable. When a goal is resumed, it is returned to the FK from the BK.

Finally, we assume a failure-driven approach to learning and reasoning (Kolodner, 1987; Schank, 1982; Schank & Owens, 1987; Sussman, 1975; Van Lehn, 1991), which concentrates on mistakes, unexpected successes, surprises, and impasses to indicate when attention is appropriate. A *failure* is defined to be a computational outcome other than what is expected or a lack of some outcome. If the system analyzes some input incorrectly, given some criteria or feedback, then a failure has occurred. This standard failure will be termed a *mistake*. Moreover, if the system expects that it will not be able to compute any answer or the correct answer, but it does nonetheless, then another kind of failure exists, called an *unexpected success*. Alternatively, if the system has no expectation, yet an event occurs which should have been expected, then a *surprise* exists. Finally, an *impasse* such that no solution can be generated is also considered a failure by definition. This paper will not claim that all learning is failure-driven (see Jones & Van Lehn, 1991 and Van Lehn, 1991 for an alternative view), but rather that failures guarantee that something worth learning exists; whereas

success may or may not provide learning opportunities.

Given such assumptions, the cognitive tasks of reasoning and learning have great parallels in Meta-XP theory. So first, this section presents a model of reasoning with its content and process theory. Section 4 will give the same accounting for the learning task.

### 3.2 Process Theory of Understanding

*"The general idea of failure-based understanding is that examining how we make comparisons between our expectations and what actually occurs is the key to our knowledge of the understanding process itself."*

--Schank & Owens (1987).

Like the operational definition of learning defined in section 1, reasoning is also viewed as failure-driven and subject to strategy selection. Thus, the reasoning task can be operationalized in general terms as follows: Given some input from the world (e.g., preprocessed perceptual input or text from a story), a current context, including contextual goals, and BK, if there exists an anomaly in the input, choose a reasoning algorithm to resolve the anomaly. As with the model of learning, the reasoning is a type of meta-reasoning, since the top level of computation concerns the choice of a reasoning algorithm, rather than the choice of a solution operator.<sup>7</sup>

Now if no unusual input to the system exists, then no significant resources will be expended on cognition. In general, reasoning may be either an understanding process or a problem-solving process, so in the absence of interesting input, an understander will skim its data; lacking interesting input, a problem solver simply acts reactively, or from habit. In such situations there is no deliber-

---

7. This multiple levels of reasoning is consistent with the approach of Stefik's (1981) MOLGEN system, in which a plane of reasoning exists in both the design plane (the reasoning task in their domains) and the meta-plane (the task of choosing an operator in the design plane). As a result of this division, to choose a reasoning strategy the system should understand and model of its own algorithms. Though consistent with Stefik, however, Meta-AQUA does not have separate planes of computation.

ation. With interesting input, however, a reasoner should select and execute a strategy, thus generating some response that resolves the anomaly that sparked the interest. Subsequently, the resolution is verified by some means chosen by the reasoner. In this formulation, then, reasoning is basically a variant of the generate-and-test paradigm, with the enhancement of a front-end identification phase to detect the anomalous, or otherwise interesting, input (see figure 3).

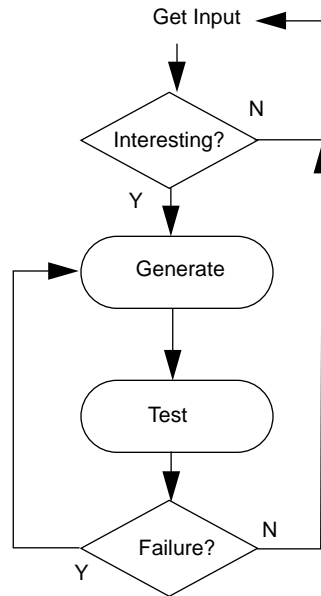


Figure 3: Augmented Generate-and-Test Cycle

The work presented here has concentrated on developing the details of reasoning in the form of understanding. A story-understanding task will thus be the processing domain with which to test our theory of introspection and learning. In particular this research develops an explicit, if somewhat simplified, model of the processing performed by an implementation of question-driven story understanding called AQUA<sup>8</sup> (Ram, 1989, 1990, 1991, 1993). In the AQUA system, understanding involves building causal explanations of the input, which provide conceptual coherence by

---

8. AQUA stands for Asking Questions and Understanding Answers.

incorporating pieces of previous input. The understander simply skims a story by instantiating schemas to fit a given input and tying them into the previous concepts of the story, unless the input is anomalous. If an anomalous situation is identified, then the reasoner must explain the input by elaborating it beyond simple schema instantiation. Thus, the understander must choose a method to generate an explanation, then select a method to test the veracity of the explanation. With respect to figure 3, generate and test correspond with constructing hypothetical explanations and verifying hypotheses, respectively.

For example, AQUA might process a story about a Syrian teenager in Lebanon named Hamida.<sup>9</sup> While processing the story, AQUA constructs a model of the character and the actions involved in the story. If the story reveals that Hamida drives a car full of explosives into an Israeli military post, killing herself and a number of soldiers, then an anomaly exists that must be explained to fully understand the story. The event is anomalous because the model of Hamida constructed before the point of her suicide was one of a normal teenager. A conflict occurs as a result of trying to unify the picture of Hamida as a typical teenage girl, assumed to be happy, with the picture of her as an individual likely to commit suicide, and thus apt to be depressed.

To explain the incongruity, the system must analyze the anomaly. AQUA accomplishes this by consulting a decision-model (Ram, 1990) that describes the planning process an agent goes through when considering the choice of actions to be performed in the world. The objective of the analysis is to refine the nature of the anomaly and to identify the parts of the story that bear on the anomaly, so as to more clearly ascertain what needs to be explained in order to resolve the anomaly. The analysis of the story would thus yield the facts that Hamida was not depressed, yet at the same time she performed an act that resulted in the loss of her own life. This situation is certainly anomalous because the decision model claims that people value the goal of preserving their own life above

---

9. This example is adapted from story S1 presented in Ram, 1990.

other goals that they possess.

Following this result, AQUA poses a series of questions about the anomaly and the context of the story surrounding the anomaly. In this case AQUA asks what would cause a girl to carry out an action she knew would result in her own death. If this question can be answered, then the anomaly would likely be resolved, and the story would be considered understood. AQUA then uses the only explanation method it knows, which is explanation application.<sup>10</sup>

AQUA can generate two types of explanations. *Physical explanations* give a causal account of events according to a model of the way things work in the world, whereas *volitional explanations* give a causal account of why people perform the acts they do in the world. The former links physical events, such as the death of individuals, with causes, such as the detonation of bombs, the role of fuses, and so forth. The latter links the actions of agents in a story to their goals and beliefs, thus providing a motivation of the character. In this scenario, AQUA retrieves and instantiates a religious-fanatic explanation, which produces expectations in the story. It can either look for verification of the explanation by tying it into the story, or it can suspend the explanation till later. The explanation is then verified when later sentences in the story confirm the hypothesis.

The model of understanding used in this thesis is a modification of the reasoning method used by AQUA. Figure 4 diagrams the processes that could produce the understanding of Hamida's story above. First the system performs simple anomaly detection. An anomaly is signaled when either the input conflicts with known facts in the BK, or when the system is otherwise unable to successfully incorporate the representation of the input into the current story model in the FK. An explanation phase then attempts to resolve the anomaly by constructing a causal account of the input with respect to both the story and the system's knowledge. The resulting hypothesis is then tested

---

10. This algorithm will be outlined in section 4.2, "Process Model of Learning," starting on page 36

for degree of fit or believability.

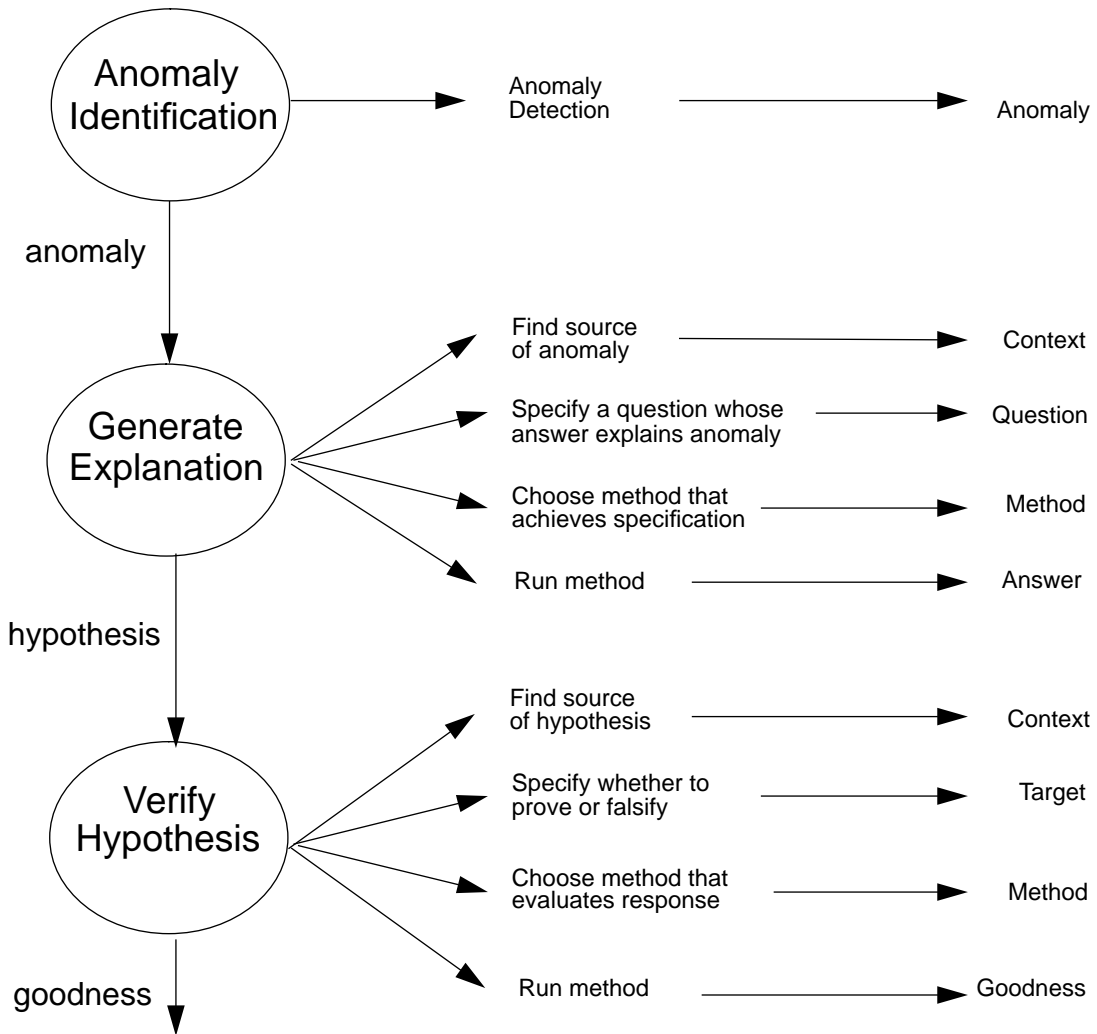


Figure 4: Question-Driven Understanding

Like the outline of the learning algorithm in the introduction, both the generate and the test components of understanding have four steps (see figure 4). In essence, given some anomalous state the reasoner encounters, if it is to explain the anomaly, and thus understand the story, it must

answer:

- How did the anomaly occur?
- What needs to be explained?
- How can I explain this?

Subsequently it will:

- Resolve the anomaly by explaining it.

The initial and most important step is to elaborate the anomaly in order to provide a focus in the story that is relevant to determining what occurred. The reasoner also refines the anomaly in such a way that a specific question can be posed. Since the specification of the explanation phase must be more precise than simply "explain the anomaly," it does little good, given some anomaly, to simply ask what the reason is for the anomaly. Although it may be clear that some representation for a character like Hamida indicates that she is a normal person, that a later representation of her is a suicidal person, and that the two representations will not unify in the program internals, a better characterization of the anomaly provides specific circumstances, including motivations, states, goals, and beliefs, in terms of a model of normative decisions and a model of the current story that point to possible locations of the anomaly. Moreover, by providing a story context a system avoids much search, since the context should contain only the pertinent details known so far. A good programmer can set up the anomalies that its system knows about in such a way that resolution is all but guaranteed. It is better to have some process that attempts to focus the anomaly so that unusual conditions not envisioned by the programmer can also be addressed.

Given such detail, the function of the next step is to provide a set of questions that represents gaps

in the model of the story with respect to the anomaly. Any such question can be viewed as a *knowledge goal* (Ram, 1989, 1991; Ram & Hunter, 1992), since it specifies the knowledge states that, if achieved, would provide coherence to both the story and what the system knows (its BK). Following this specification the system can pick an explanation method that will answer these questions. Though AQUA itself does not choose a method, but has only a single procedure, any number of abductive methods will suffice instead (Ram & Leake, 1991). Once a strategy is settled on, the program can generate the explanation.

The first step in explanation generation is similar to the blame assignment step in learning, the second is goal specification, and the third is strategy selection. After performing these steps, the reasoner can simply execute the reasoning method. Like the learning model of this paper, which offers no new learning algorithms and instead presents a method of choosing between a number of extant strategies, the reasoning model offers no new reasoning algorithms. Depending upon the given situation, a system may choose from case-based reasoning, analogy, explanation application, or any number of reasoning methods for generation. To perform a test of the resulting hypothesis, a reasoner may devise an experiment, ask someone, or simply wait, in the hope that the answer will be provided by future input.<sup>11</sup>

To verify the hypothesized explanation, the verification process makes a similar four-step analysis. The first step, however, that of finding the source of the hypothesis, is known to follow in sequence from the generation process.<sup>12</sup> Step two is to determine whether to attempt to prove or disprove the hypothesis. Given a target approach, the system then needs to choose an algorithm best suited to achieving the goal. Once the algorithm has been selected, the hypothesis can then be evaluated.

---

11. The structure of this strategy selection method allows for easy incorporation of additional algorithms.

12. Yet in instances where a hypothesis is not self-generated, but provided to the reasoner as input, step one would indeed require significant computation.

With this model, the operationalized statement of the task of understanding is as follows: Given some input from the story, the system's current foreground knowledge ( $FK_1$ ), including contextual goals and a current representation of the story, and the system's background knowledge (BK), if an anomaly exists in the input, choose a reasoning algorithm to explain the anomaly, else incorporate the input into  $FK_1$ . Output a new representation of the story ( $FK_2$ ), including a representation of the reasoning that produced it, that has no anomaly and is coherent with respect to the BK.

### 3.3 Content Theory of Understanding

Early research by Davis (1980) argued for the importance of metaknowledge, especially in the form of declarative search-control rules. Although many in the artificial intelligence community have recognized the necessity of reasoning about one's own beliefs (e.g., Davis & Buchanan, 1977; Maes, 1988), and some have even called it introspection (e.g., Konolige, 1988), few have both modeled and represented the processes that *generates* beliefs,<sup>13</sup> and made them available to the reasoner itself. This section develops a brief content theory of understanding in order to formalize knowledge about the reasoning itself.

A Meta-Explanation Pattern (Meta-XP) is an explanation of an explanation. Whereas an Explanation Pattern (XP) is a causal structure that explains a physical or mental state by presenting the chain of physical or mental events that results in such states in the world or in the mind (Ram, 1989; Schank, 1986; Schank & Leake, 1990), a Meta-XP is an explanation of how or why an XP is mis-generated or otherwise fails (Ram & Cox, to appear).<sup>14</sup> There are two classes of Meta-XPs. A Trace Meta-XP (TMXP) explains how a system generates an XP about the world or itself, and an

---

13. A prominent exception is Collins, Birnbaum, Krulwich & Freed (1992) who argue that to plan effectively a system must have an explicit model of its of planning and execution processes.

14. Here the definition of a Meta-Explanation is interpreted in a narrow sense as applied to understanding tasks involving the explanation of anomalies. In general, however, a Meta-XP may be any explanation of how and why an agent reasons in any particular way, including processes other than explanation.

Introspective Meta-XP (IMXP) explains why the reasoning captured in a TMXP goes awry. This section will describe and explain the use and structure of TMXPs, while IMXPs will be described in section 4.2, “Process Model of Learning.”

Ram (1990) has developed a theory of motivational explanation based on decision models which characterize the decision process that an agent goes through in deciding to perform an action. For example, the religious-fanatic explanation for suicide bombing is a decision model describing why a bomber would choose to perform a terrorist strike in which the bomber dies. Ram's model claims that an agent first considers its goals, goal priorities, and the expected outcome of performing the action. The actor then makes a decision whether or not to enter into such a role, and if so, performs the action. Meta-XP theory extends the model to account for introspective reasoning.

Reasoning in general can be performed in a similar manner. A set of states, priorities, and the expected strategy outcome determine a reasoner's decision of a processing strategy, like the above factors determine the actor's decision to act. Based on general knowledge, current representation of the story, and any inferences that can be drawn from this knowledge, the reasoner chooses a particular reasoning strategy. Once executed, a strategy may produce further reasoning requiring additional strategy decisions.

These decisions are chained into threads of reasoning such that each one initiates the goal that drives the next. Though the chains can vary widely, in the task of question-driven story understanding, the chains take the form shown in figure 5: Anomaly Identification → Generate Explanation → Verify Hypothesis. Note that since the explanation generation phase produces a hypothesis and the verification phase produces a measure of goodness, if the hypothesis has been confirmed with a sufficiently high confidence, then the overall product of the understanding process has been a sound explanation. Alternatively, if the explanation has been disconfirmed, then a later failure iden-

tification phase should generate the question "Why did the explanation fail?" This knowledge goal triggers the learning process.

The understanding process is recursive in nature. For example, if a hypothesis generates a new question, then the reasoner will spawn a recursive regeneration of the sequence because an unanswered question is anomalous. Like physical explanations that explain how objects work in the physical world, and volitional explanations that explain why agents perform various acts in the world, introspective explanations explain how and why conclusions are drawn by the reasoner; they explain events in the mental world.

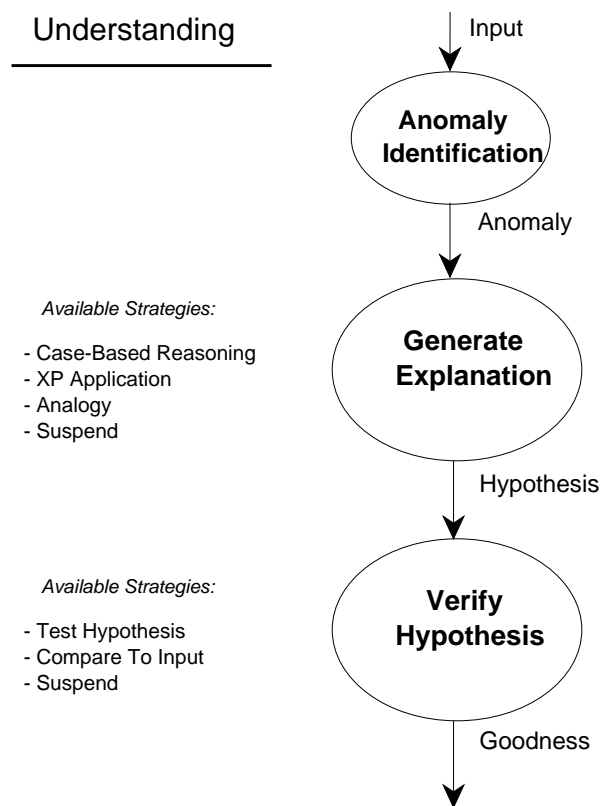


Figure 5: Phases of Understanding

When insufficient knowledge exists on which to base a decision, a useful strategy is to simply defer

making the decision. The reasoning task is suspended and later continued if and when the requisite knowledge appears. This is a form of opportunistic reasoning (Birnbaum & Collins, 1984; Hammond, 1988; Hayes-Roth & Hayes-Roth, 1979; Ram, 1989). Meta-XPs are able represent chains of reasoning that follow from opportunistic reasoning as well as uninterrupted decisions.

A Trace Meta-XP, representing the trace of the reasoning process, is a chain of Decide-Compute-Nodes (D-C-Nodes). A non-recursive single instance of explanation would be a chain of three D-C-Nodes, one for each phase in the anomaly-identification/generate-explanation/verify-hypothesis sequence.<sup>15</sup> These nodes (see figure 6) record the processes that formulate the knowledge goals of a system, together with the reasons for and the results and side-effects of performing such mental actions. The trace of reasoning is similar to a derivational analogy trace as described by Carbonell (1986) and Veloso and Carbonell (1991). A Trace Meta-XP is a specific explanation of why a reasoner chooses a particular reasoning method and what results from the strategy. Like an XP, the Meta-XP can be a general structure applied to a wide range of contexts, or a specific instantiation that records a particular thought process. One distinguishing property of Trace Meta-XPs is that a decision at one stage is often based on features in previous stages. For example, the decision of how to verify a hypothesis may be based on knowledge used to construct the hypothesis initially. This property, deciding based on previous knowledge, is particularly true of learning, which, by definition, is based on prior processing.

---

15. Note that in most of this work the initial phase of anomaly identification is simplified. Rather than considering all four steps represented in a D-C-Node, the algorithm skips the input analysis step, posts a goal to interpret the input, and then uses only a single strategy as outlined in the text. The result is a signal whether or not an anomaly exists together with the anomaly's cause. See, for example, Figure 4: "Question-Driven Understanding" on page 18.

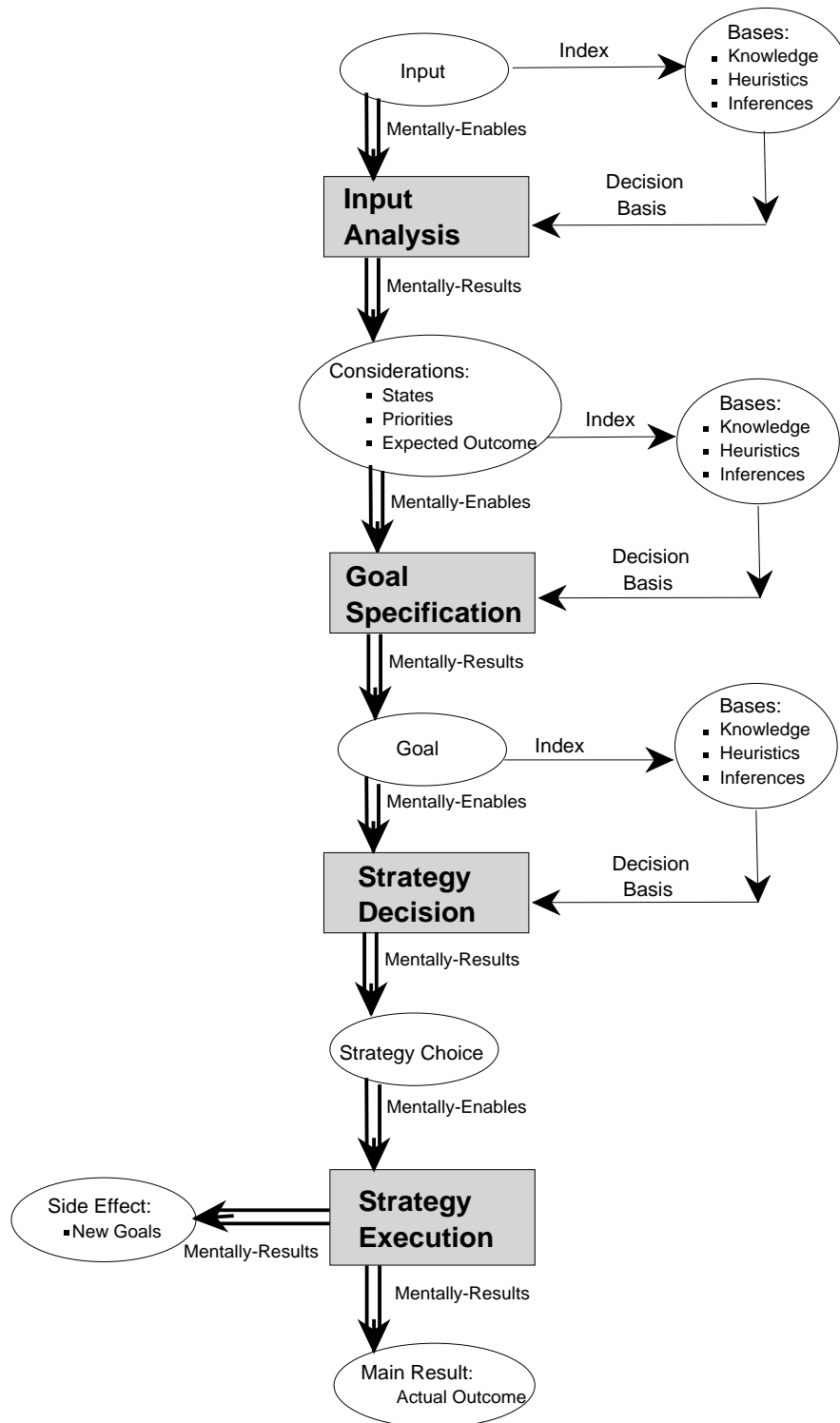


Figure 6: Decide-Compute-Node

### 3.4 Taxonomy of Reasoning Failures

Given the first assumption of figure 2, that reasoning is the goal-directed processing of a given input using the reasoner's knowledge, then only a limited number of classes of faults can be responsible for a given failure: the reasoning failure can originate in either the reasoner's goals, its processing strategies, the input, or the domain knowledge (see table 1). Given the third assumption of figure 2, that knowledge is memory-based and that the indexing problem is a serious issue, then the organization of suspended goals (indexes), processing strategy associations (heuristics), or the organization (indexes) of the domain knowledge may also be to blame. If one of these categories is responsible for an error, the item corresponding to the category is either absent or incorrect. If an item is correct, then that category contributes nothing to the failure.<sup>16</sup>

The most basic type of failure occurs when the system's domain knowledge is at fault. A classic domain theory, such as the cup domain, presents the rules, concepts, and relations involved in a particular self-contained knowledge system. A domain theory is considered incomplete<sup>17</sup> if pieces of the knowledge base are missing (Novel Situation). In rule-based systems incompleteness occurs when a rule or an antecedent of a rule is missing, while frame-based systems are incomplete when concepts or attributes are missing. A domain theory is considered incorrect if there are pieces of the knowledge base present that should not be (Incorrect Domain Knowledge). In rule-based systems this occurs when an extra rule or antecedent of a rule is present, while frame-based systems are inconsistent when concepts or attributes are present that should not be. Domain theories are overly specific when they are missing some rules or when they possess extra antecedents (in a

---

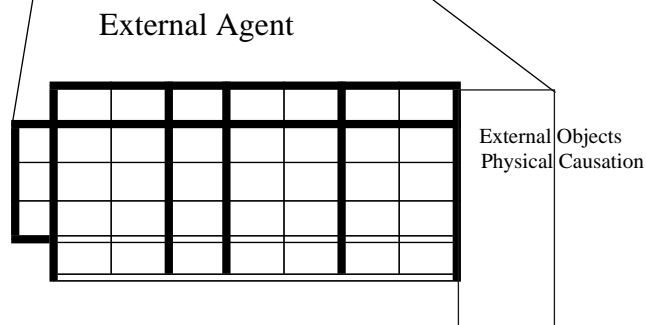
16. One of the targets of this research is to produce representations for all of the cells of table 1. At present, the input column, the domain knowledge, and knowledge selection columns have explicit Meta-XP representations, many of which will be shown in section 4.3, "Content Theory of Introspective Explanations," starting on page 43. Additionally, both the Missing Goal and the Forgotten Goal cells of the table are represented (see Cox and Ram, 1992b). The representations of the various cells are chained into composite structures that capture typical failure patterns such as the example in section 4.1, "Implementation and Example," starting on page 33.

frame system this entails missing types or extra preconditions), since a missing rule would include an example left out, whereas an extra antecedent disallows an example that should be included. Domain theories are overly general when they are missing antecedents or when they possess extra rules (in a frame system this entails missing preconditions or extra types), since an extra rule included an example that should be left out, while the inclusion of a missing antecedent rejects an example that was not left out (Mooney & Ourston, 1991).

**Table 1: Dimensions of Failure<sup>a</sup>**

	Goal Generation	Goal Selection	Input	Processing Strategy	Strategy Selection	Domain Knowledge	Knowledge Selection
Absent	Missing Goal	Forgotten Goal	Missing Input	Missing Behavior	Missing Heuristic	Novel Situation	Missing Association
Wrong	Poor Goal	Poor Priority	Noise	Flawed Behavior	Flawed Heuristic	Incorrect Domain Knowledge	Erroneous Association
Right	Correct Goal	Correct Association	Correct Input	Correct Behavior	Correct Choice	Correct Knowledge	Correct Association
	Resource Scheduling	Opportunism	Perception	Action	Control	Domain Theory	Memory

a. For additional details concerning this table see Cox, 1992 and Ram, Cox & Narayanan, 1992.



17. Note that the use of incompleteness as a logical term is different. A logically incomplete domain theory is one in which a positive example of a category cannot be proven. This may occur if an extra antecedent on a rule exists, not just when rules are missing. A missing antecedent does not itself lead to logically incomplete theories.

Given the indexing problem, however, there may be failures, not because there was no piece of knowledge to interpret an input, but rather because the knowledge was not indexed so that it could be retrieved at the right time to use the knowledge. This error (Missing Association) occurs when the index is overly specific, since it does not match the cues in the context at retrieval time. In addition, failures can also occur when knowledge is brought to bear at inappropriate times. This type of failure (Erroneous Association) occurs when an index matches a cue that it should not. The index is then overly general.

The processing strategy and goals columns can be thought of in much the same way as the domain knowledge column; in addition to being either missing or wrong, they too have a organizational component and are thus subject to the indexing problem. As described earlier, opportunistic reasoners suspend goals that cannot currently be achieved, indexing them in memory where they can be retrieved at a later time. Thus a failure may occur, not because the reasoner never generated the goal to accomplish a particular state, but rather because the goal was stored using indexes that did not match the cues present in the input at a later retrieval time (cf., stranded motorist example of section 1, "INTRODUCTION").

Reasoning strategies are applied only if they are selected using some heuristic that determines they are applicable. Thus the heuristics can be thought of as "indexes." So a failure may occur, not because it does not have the strategy with which to process the input, but rather because it does not have the specific heuristic to signal the strategy's application, or because another heuristic selects a competitive strategy.

The most complicated column is the one representing the input to the system. It is the input that constitutes the interface between internal and external worlds. As suggested by the sub-table, if one allows interaction with other agents in the world, then blame may be associated with the goals,

strategies, input and knowledge of other agents. Thus, for example, noise in the input may actually be due to the deception of opponents caused by conflicting goals of the external agent. This con-  
volution makes blame assignment exceedingly difficult; however, if one is to categorize open-  
world situations, then this source of blame cannot be ignored. The input column is most interesting  
in that it involves an input, the perception of the input, and the interpretation of the perception of  
the input. The treatment in the following section, however, ignores such discriminations, instead it  
concentrates on the analysis of the main table alone.

For each dimension represented by a column in the table, a general characterization of it appears  
at the bottom of the column. For each of the columns: goal generation, processing strategy and  
domain knowledge, a natural dualism is present in its interpretation. For example, strategies repre-  
sent both mental and physical actions. Thus there exist mental actions such as operators for mental  
arithmetic (e.g., integrate by parts in the calculus domain of Mitchell, Utgoff & Banerji's (1983)  
LEX program) as well as physical actions like robot navigation schemas (e.g., turn-left). For each  
type of action, associated heuristics are used by a reasoner to choose when to apply the action. In  
a similar fashion, there is a physical and mental manifestation of goals and domain theories. Thus,  
an agent can have mental reasoning goals, such as "remember where I parked the car," and can also  
have goals to achieve states in the world, like "be at my car's location."<sup>18</sup> Likewise an intelligent  
agent can have knowledge about the world as well as self-knowledge. Although these nuances are  
important distinctions to observe, the preliminary treatment presented here will treat each cell in  
the matrix simply.

A number of invariants and constraints exist within the table. For instance, it is not possible to have  
both a correct association and an erroneous association. If the correct cell is enabled, then neither  
of the other two cells in the column can coexist. It is also impossible for an item in some column

---

18. Note that the former is a subgoal of the latter.

to be both missing and wrong. These constraints exist as a result of simple logic. Other relations hold because of the semantics of the failures that the table captures. For example, it is not possible for an Erroneous Association to exist without either a Novel Situation or a Missing Association also present, because an Erroneous Association signals an expectation failure: something was retrieved that should not have been. Something else should have been retrieved instead. Thus, something is either not present in the domain knowledge (and thus cannot be retrieved) or it was present, but no association was present with which to retrieve it. Part of the research associated with this taxonomy, then, is to ascertain those relations that exist among the cells of the table and to use these to constrain inference in the system.

Another aspect of the failure taxonomy that must be addressed is that the symptoms for each type are ambiguous. For example it is difficult in many situations to distinguish between noise in the input and a novel situation. If an astronomer believes she just experienced a never-before-observed celestial event, then how does the scientist determine if it was a truly novel occurrence or simply a glitch in the measurement device? The event could be disambiguated if other telescopes witness the same event. If the telescope has had recent malfunctions, however, then the scientist may have a bias to falsify the hypothesis, rather than to corroborate it. Developing heuristics to determine such judgements are an important part of the research that will accompany this thesis.

## **4 LEARNING MODEL**

Much of the focus of the research has been on the taxonomy of reasoning failures and the associated strategies that are useful in learning from such failure situations. The implementation of the taxonomy seeks to exploit the matching of descriptions of failure (Trace Meta-XPs) with descriptions of solutions (Introspective Meta-XPs). Rather than provide the mapping directly, which is dif-

difficult, the mapping is done through the formulation of learning goals, which are associated with learning plans or strategies that help achieve those goals (Hunter, 1990; Ram and Hunter, 1992). As outlined by the functional justifications presented in section 4.2.2, “Process Theory Functional Arguments,” a number of benefits result from the mediation of the mapping by learning goals. However, one can describe the effective mapping from failure situations to learning strategies using the following categories of reasoning failures along with the corresponding types of learning that need to be performed.

- *Missing Association*: The reasoner may have an applicable knowledge structure to deal with a situation, but it may not be indexed in memory such that it can be retrieved using the particular cues provided by the context. In this case the system must add a new index, or generalize an existing index based on the context (Cox and Ram, 1991; Hammond, 1989; Ram, 1993).
- *Erroneous Association*: If on the other hand, the reasoner retrieves a structure that later proves inappropriate, it must specialize the indexes to this structure so the retrieval will not recur in similar situations (Cox and Ram, 1991; Hammond, 1989; Ram, 1993).
- *Novel Situation*: A failure can arise when the reasoner does not have the appropriate knowledge structures to deal with a situation that is truly novel. In such cases, the reasoner could use a variety of learning strategies, including explanation-based generalization (DeJong and Mooney, 1986; Mitchell, Keller & Kedar-Cabelli, 1986) or explanation-based refinement (Ram, 1993), coupled with index learning (Bhatta & Ram, 1991; Hammond, 1989; Ram, 1993) to organize the new knowledge structures.
- *Incorrect Domain Knowledge*: Even if the reasoner has applicable knowledge structures, they may be incorrect or incomplete. Learning in such situations is usually incremental, and involves strategies such as elaborative question asking (Ram, 1991, 1993) applied to the reasoning chain,

and abstraction and generalization techniques (Michalski, 1991) applied to the domain knowledge.

- *Missing Input*: The system requires information or otherwise detects some information that is lacking in the givens. A question is formed to fill the gap in the current knowledge (Ram, 1989, 1991). Questions may be answered by inferring the information from the background knowledge, actively pursuing the answer by querying data bases, asking the user, and so on, or by suspending the process and opportunistically waiting for an answer in further input. When the answer arrives, the system performs standard inductive learning if the answer is not novel. If, however, the answer is unusual, the system categorizes the failure recursively according to the system's failure taxonomy and applies the learning strategies associated with the new failure. Thus, answers to questions may prompt the formation of additional questions.

- *Input Noise*: The reasoner may possess the right knowledge, have it organized in a proper manner, and use the correct reasoning methods, yet fail due to incorrect or incomplete external knowledge sources. In reasoning tasks, the blame may be due to measurement errors, obsolete data, missing data, or explicit deception by another agent. The solution is to learn the conditions under which knowledge sources are reliable and the kinds of data that are necessary in a given situation (Booker, Goldberg & Holland, 1989).

- *Incorrect Reasoning Choice*: This failure type occurs when the reasoner has an appropriate knowledge structure to reason with and index to the structure in memory, but incorrectly chooses the wrong knowledge because the reasoning method it decided to use turned out to be inappropriate or inapplicable. An analysis of the choice of reasoning methods results in learning control strategies designed to modify the heuristics used in this choice (Mitchell et al., 1983; Sleeman, Langley & Mitchell, 1984).

The current implementation of Meta-AQUA focuses mainly on the first four types of errors listed

above since the main theoretical emphasis is on the integrated introspective learning architecture. Meta-AQUA is currently being extended to deal with the other three types of enumerated failures.

## 4.1 Implementation and Example

This research implements an introspective version of AQUA, called Meta-AQUA. AQUA is a question-driven story understanding system that learns about Middle Eastern terrorist activities. Its performance task is to "understand" the story by building causal explanations that link the individual events into a coherent whole. Meta-AQUA adds introspective reasoning and learning using Meta-XP structures. Meta-AQUA is programmed in Symbolics Common LISP under Genera Version 8.1.1 on a Symbolics MacIvory-3 LISP processor embedded in a Macintosh II computer. The LISP source files take up 429,731 bytes of disk space.

Unlike AQUA, Meta-AQUA does not actually parse the sentences; since this research does not deal with the natural language understanding problem, Meta-AQUA assumes that input sentences are already represented conceptually. The BK used in the current implementation consists of a frame-based conceptual hierarchy, a case library of past episodes, and an indexed collection of XPs. To illustrate the type of introspection Meta-AQUA performs and the type of learning that results, consider the passage in figure 7.

- 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: At this point 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 7: The Drug Bust Story

A number of inferences can be made from this story, many of which may be incorrect, depending

on the knowledge of the reader. Meta-AQUA's knowledge includes general facts about dogs and sniffing, and it knows that dogs bark when threatened, but it has no knowledge of police drug dogs in particular. It also knows of past terrorist smuggling cases, but has never seen a case of drug interdiction. Nonetheless, the program is able to recover and learn from the erroneous inferences this story generates.

S1 produces no inferences other than the observation that sniffing is a normal event in the life of a dog.

S2 produces an anomaly, however, because the system's definition of "bark" specifies that the object of the bark is animate. In this example, the program (incorrectly) believes that dogs bark only when threatened by animate objects. Since luggage is inanimate, a contradiction exists, leading to a constraint anomaly. This anomaly causes the understander to ask why the dog barked at an inanimate object. Because the BK of the program has only one relevant abstract Explanation Pattern telling why dogs bark, it is able to produce only one instantiated explanation: the luggage somehow threatened the dog.

S3 asserts an arrest scene which reminds Meta-AQUA of a prior incident of weapons smuggling by terrorists. The system then infers the existence of a smuggling bust that includes detection, confiscation, and arrest scenes. Because baggage searches are the only detection method the system knows, the sniffing event remains unconnected to the rest of the story.

Finally, S4 causes the question generated by S2 "Why did the dog bark?" to be retrieved, and the understanding task is resumed. Instead of revealing the anticipated threatening situation, S4 produces another hypothesis: the dog barked because it knows drugs are in the luggage. The program prefers the explanation given by S4 over the earlier one because the explanation provides greater coherence to the story representation. However now the system is confronted by the fact that it

had predicted one explanation to hold, but another explanation proved correct instead. It characterizes the reasoning error as one in which there is 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). Using this characterization as an index, the system retrieves an introspective explanation to apply to the trace of the reasoning that produced the errors.

The explanation helps in understanding why the prior reasoning failed, in determining what needs to be learned in order to revise the system's BK, and in selecting a learning algorithm to make the revision. The explanation represents a common failure pattern such that a restriction on the type definitions in conceptual memory causes the system to believe that an input is anomalous. Actually though, the situation is novel in that it has not been experienced before by the system. Thus, the system has no explanation for the anomaly, but instead attempts to apply an explanation that it already has for a seemingly analogous situation. But this analogy fails.

Applying this introspective explanation to the failure trace allows Meta-AQUA to relax its constraint on objects of dog barking to include inanimate as well as animate objects. Thus, the system performs abstraction on the constraint, raising it to physical-object. Furthermore, the new explanation for the dog barking at the luggage is generalized via Explanation-Based Generalization (EBG) (DeJong & Mooney, 1986; Mitchell et al., 1986) to an abstract XP which asserts that dogs bark at containers when they detect contraband. This new XP is then compared to the old XP, and they are indexed so that the former applies when dogs bark at inanimate objects, whereas the latter applies when dogs bark at animate objects.

Though the program is directly provided an explanation that links the story together, Meta-AQUA performs more than mere rote learning. It learns to avoid the mistakes made during the processing

of the story. The application of Meta-XPs allows the system to use the appropriate learning strategy (or multiple strategies) to learn exactly that which the system needs to know to process similar situations in the future correctly. A subsequent story in which a police dog is used to find a marijuana plant in a trash bin within a suspect's home produces no errors.

## **4.2 Process Model of Learning**

Meta-AQUA records its reasoning during the task of story understanding in a trace structure similar to the derivational analogy traces of the PRODIGY system (Veloso & Carbonell, 1991). These knowledge structures contain representations for each of the reasoning phases: anomaly identification, hypothesis formation, and verification. For each phase the structure records the considerations that prompted the phase, the bases for making a reasoning strategy decision, and the result of such strategy execution. After reasoning completes an inference chain, the reasoning trace is passed to a learning process. The learning must then check for failures, explain and learn from the failure, and in the ideal model, it should then verify that the learning was reasonable (see figure 8). When failures occur, the system needs to explain why the failure occurred by applying an introspective explanation to the trace. This explanation aids in blame assignment and in determining what to learn. An Introspective Meta-XP (IMXP) provides a set of learning goals that are designed to modify the BK of the system, including the organization (indexes) of that knowledge, in order to reduce the likelihood of the error from recurring.

The main control algorithm of Meta-AQUA used for introspective (second-order) reasoning is essentially the same as the XP-application control algorithm used in explanatory (first-order) reasoning in the AQUA and SWALE (Kass, Leake, & Owens, 1986; Schank & Leake, 1990) systems. To describe how the introspective reasoning works, some background information on XP-application follows.

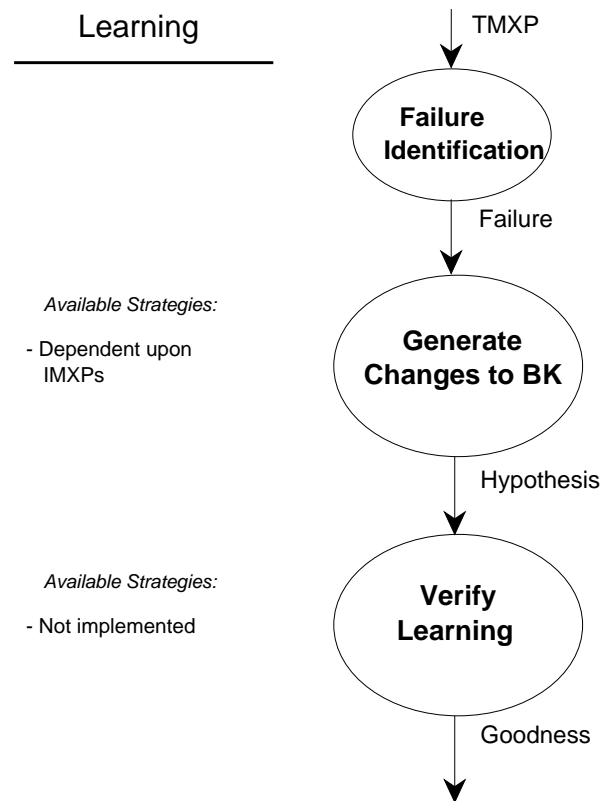


Figure 8: Phases of Learning

Explanation Patterns (XPs) are similar to justification trees, in that they link antecedent conditions to their consequences. The XP is essentially a directed graph of concepts connected with RESULTS, ENABLES and INITIATES links. A RESULTS link connects a process with a state, while an ENABLES link connects a precondition state to a process. An INITIATES link connects two states. The set of sink nodes in the graph is called the PRE-XP-NODES. These nodes represent what must be present in the current situation for the XP to apply. One distinguished node in this set is called the EXPLAINS node. It is bound to the concept which is being explained. Source nodes are termed XP-ASSERTED-NODES. All other nodes are INTERNAL-XP-NODES (Ram, 1990, 1991, 1993).

For an XP to apply to a given situation, all PRE-XP-NODES must be in the current set of beliefs.

If they are not, then the explanation is not appropriate to the situation. If the structure is not rejected, then all XP-ASSERTED-NODES are checked. For each XP-ASSERTED node verified, all INTERNAL-NODES connected to it are verified. If all XP-ASSERTED-NODES can be verified, then the entire explanation is verified. Gaps in the explanation occur when one or more XP-ASSERTED-NODES remain unverified. Each gap results in a question, which provides the system with a focus for reasoning and learning, and limits the inferences pursued by the system. Given this methodology, the algorithm for explaining and learning from a reasoning failure works much the same way. Figure 9 outlines a similar control algorithm for an introspective multistrategy learner.

The identification of blame during the learning phase is analogous to the method used in AQUA or SWALE to explain anomalies in story inputs. Instead of taking as input a conceptual representation of events in the world and outputting an explanation of the anomaly, however, the blame assignment process in Meta-AQUA takes as input a conceptual representation of the reasoning performed in explaining an event in the world and outputs an explanation of the reasoning failure. Just as the XP application algorithm can be applied to the world events, the algorithm in figure 9 can be applied to the mental events, using Meta-XPs with a single level of recursion. A characterization of the reasoning failure is used as an index to retrieve an abstract IMXP. This structure is then bound with the trace of the reasoning to produce a variablized token. The sink nodes (PRE-XP-NODES) in the structure are then checked to see if they are consistent with the current representation of the reasoning that produced an understanding the story. If they all can be verified then the Meta-XP applies to the situation. If any are rejected, then the explanation is rejected. If any nodes are neither confirmed or rejected, then a question is posed on the node. If the question cannot be answered, then the introspection is suspended, the reasoning is indexed in memory and the process is suspended. Later reasoning can opportunistically resume the process in the future.

## 0. Perform and Record Reasoning in TMXP

### 1. Failure Detection on Reasoning Trace

### 2. If Failure Then

#### Learn from Mistake:

- Blame Assignment

Compute index as characterization of failure

Retrieve Introspective Meta-XP

Apply IMXP to trace of reasoning in TMXP

If Successful XP-Application then

    Check XP-ASSERTED-NODES

    If one or more nodes not believed then

        Introspective questioning

        GOTO step 0

    Else GOTO step 0

- Post Learning Goals

- Choose Learning Algorithm(s)

- Apply Learning Algorithm(s)

### (3. If Learning Then

#### Evaluate Learning)

Figure 9: Introspective Multistrategy Learning Algorithm

Once an IMXP is retrieved and successfully applied to the trace of failed reasoning provided by some TMXP, the system must generate the learning as outlined in the introduction. Given some anomalous state the reasoner encounters, then, the three tasks of performing blame assignment, formulating a learning goal, and choosing a learning strategy is equivalent to answering the following three questions:

- How did I get here?
- Where should I go to get out of here?
- How can I get there?

Subsequently it will:

- Go there as planned.

The first problem is to explain how the reasoner got itself in such a state of failure. Given such an understanding, it then needs to figure out what direction to take to assure that this failure does not repeat. Once a direction is chosen, then a method for getting to the location (goal) is determined. Until now, the language describing the process of learning has been rather vague. A more specific definition can now be provided.

#### **4.2.1 Operationalized Phases of Learning**

As the introduction argues in Section 1, there are three fundamental learning problems, all of which must be addressed for effective learning in open-world scenarios. They are the credit or blame assignment problem (Birnbaum, Collins, Freed, & Krulwich, 1990; Freed, Krulwich, Birnbaum & Collins, 1992; Minsky, 1963; Stroulia, Shankar, Goel & Penberthy, 1992; Weintraub, 1991), deciding what to learn (Hunter, 1989, 1990; Keller, 1986; Krulwich, 1991; Leake & Ram, to appear; Ram, 1991; Ram & Hunter, 1992), and the strategy selection problem (Cox & Ram, 1991; Ram & Cox, to appear; Michalski, 1991). The approach this research has taken in solving these problems is summarized by the following operational definitions.

- *Blame assignment*: Take as input a trace of the mental and physical events that led to or preceded a failure; produce as output an explanation of how and why the failure occurred, in terms of the causal factors responsible for the failure. The input TMXP describes how results or conclusions

were produced by specifying the prior causal chain (both of mental and physical states and events). Then retrieve an abstract IMXP from memory and apply it to the TMXP in order to produce a specific description of why these conclusions were wrong or inappropriate. This instantiation specifies the causal links that would have been responsible for a correct conclusion, and enumerates the difference between the two chains and two conclusions (what was produced and what should have been produced). Output the instantiated explanations.

- *Deciding what to learn:* Take as input a causal explanation of how and why a failure occurred; generate as output a set of learning goals which, if achieved, can reduce the likelihood of the failure repeating. Include with the output, both tentative goal-dependencies and priority orderings on the goals.

- *Strategy selection:* Take as input a trace of how and why a failure occurred and a set of learning goals along with their dependencies; choose as output a set of learning strategies to apply in order to accomplish the goals along with updated orderings on the set of goals. These learning strategies are organized as plans to accomplish the goals. The plans are sequences of steps representing calls to specific learning algorithms such as EBG or index learning. Instantiate, then execute the plans.

Blame assignment is a matter of determining what was responsible for a given failure. Thus the function of blame assignment in the Meta-AQUA system is to identify which of the possibilities and their interactions could have led to the reasoning failure (see table 1).

#### **4.2.2 Process Theory Functional Arguments**

Some of the advantages of implementing the mechanisms between the blame assignment and strategy selection stages by way of learning goals follow. These advantages are also common to standard planning paradigms.

- *Allows decoupling of many-to-many relationships.* For a given failure there may be more than one algorithm which needs to be applied for successful learning. Conversely, a given algorithm may apply to many different types of failures. The direct mapping of the possibilities (from blame to choosing an algorithm) is more difficult and less flexible than the use of learning goals.
- *Allows an opportunistic approach to solving learning problems.* It is not always possible for sufficient resources and/or knowledge to be available to perform learning when a system realizes that it needs to learn. At the time this condition occurs the system can index the learning goal in memory so that it can be retrieved at a later time when these requirements become available.
- *Allows chaining, composition, and optimization of the means by which learning goals are achieved.* In many cases there may be significant overlap between several algorithms and a number of goals. For example, two or more goals may be achieved with one algorithm, or multiple algorithms may apply to a single goal. If more than one plan applies to the achievement of a particular goal, a system should use the one that contributes to the maximal achievement of other goals with the minimal amount of resources.
- *Allows detection of dependency relationships, so that goal violations can be avoided.* It is important to recognize that when multiple items are learned from a single episode, the changes resulting from one learning algorithm may affect the knowledge structures used by another algorithm. Such dependencies destroy any implicit assumption of independence built into a given learning algorithm that is used in isolation. For example, one learning algorithm may split a concept definition into separate schemas, or otherwise modify the definition. Therefore, an indexing algorithm that uses the attributes of concepts to create indexes must necessarily follow the execution of any algorithm that changes the conceptual definition.

Another important issue in the strategy selection problem is the manner in which learning plans are

created (Hunter, 1990; Redmond, 1992). There are two approaches to this issue: a system can have either a static or a dynamic planner. A static planner simply uses the characterization of the learning goal and the context (explanations produced by the blame assignment phase) as an index into a memory of stereotypical plans. Once retrieved, a learning plan is instantiated and parameterized by the context, and then executed. More flexible and complex, the dynamic approach performs goal-subgoaling to produce plans rather than simply using canned or hand-tailored plans.

This thesis claims that to recover from failure in open-world applications, it is necessary to perform multistrategy learning. Single strategy systems are not sufficient. Furthermore, to perform multistrategy learning, one must worry about 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, dependency relations between algorithms at run-time, and alternative solutions and interactions. Treating the learner as a planner is a principled way to treat these difficulties. Many of the techniques from the planning literature, such as nonlinear and case-based methods, can be appropriated in multistrategy systems. Though Meta-AQUA is at present handling the learning problems statically, and in relatively simple situations, a dynamic planning scheme will be necessary to eventually achieve even simple levels of robustness.

### **4.3 Content Theory of Introspective Explanations**

Whereas a Trace Meta-XP explains how a failure occurred, by providing the sequence of mental events and states along with the causal linkage between them, an Introspective Meta-XP explains why the results of a chain of reasoning are wrong. The IMXP posits a causal reckoning between the events and states of the TMXP. In addition, an IMXP provides a learning goal specifying what needs to be learned. Then, given such an explanation bound to a reasoning chain, the task of the system is to select a learning strategy to reduce the likelihood of repeating the failure.

An IMXP consists of six distinctive parts:

- The IMXP type class.
- The failure type accounted for by the IMXP.
- A graph representation of the failure.
- Temporal ordering on the links of the graph.
- An ordered list of likely locations in the graph where processing errors may have occurred.
- A corresponding list of knowledge goals to be spawned for failure repair.

There are three classes of IMXPs: base, core, and composite. *Base* types constitute the blocks with which *core* IMXPs are built. We have identified six types in the base class: successful prediction, inferential expectation failure, incorporation failure, belated prediction, retrieval failure, and input failure. The core types are representations of the failure types described by the failure taxonomy, such as Erroneous Association, Novel Situation and Incomplete Domain Knowledge. Core types are combined to form *composite* IMXPs that describe situations encountered by reasoning agents, such as the example in section 4.1.

The internal graph structure of an IMXP consists of nodes, representing both mental states and mental events (processes), and the causal links between them. The nodes and links have the same semantics as those described for TMXPs at the beginning of section 4.2. The graph gives both a structural and a causal accounting of what happened and what should have happened when information was processed.

*Knowledge goals* represent what a system needs to learn (Ram, 1991, 1993; Ram & Hunter, 1992) and are spawned by the decide-what-to-learn stage. Knowledge goals help guide the learning process by suggesting strategies that would allow the system to learn the required knowledge. The Meta-AQUA system distinguishes between three types of learning goals. A *Knowledge Acquisi-*

*tion Goal* constitutes a desire for knowledge to be added to the BK. A *Knowledge Organization Goal* represents a goal to adjust the indexes that organize the system's knowledge. Using such indexes a system can efficiently retrieve appropriate structures with which an input can be understood or processed. Finally, a *Knowledge Removal Goal* is a desire to intentionally forget a piece of knowledge that has proved fruitless or actually detrimental to the function of the system. The knowledge goals spawned by an introspective examination of a reasoning failure are achieved by the use of learning plans, similar to those described by Hunter (1990) and Redmond (1992). The plans are implemented as action sequences which call various learning algorithms. Because the knowledge goals have pointers to the trace of the introspective reasoning, they have access to the TMXPs and IMXPs involved in the analysis of the failure.

#### **4.3.1 Base Class IMXPs**

Base class IMXPs represent a primitive type or component in the content theory of mental events from which traces of reasoning failures may be constructed. The goal is to enumerate a sufficient number of these basic types to cover the major kinds of reasoning failures that arise in story understanding and other tasks. The types of failures discussed in the introduction to section 4 fall into two complementary classes: commission error and omission error. Commission errors stem from reasoning which should not have been performed or knowledge which should not have been used. Omission errors originate from the lack of some reasoning or knowledge. The content theory herein contains Base IMXPs to describe both classes of failure.

We have identified two types of commission errors: *Inferential expectation failures* typify errors of projection. They occur when the reasoner expects an event to happen in a certain way, but the actual event is different or missing. *Incorporation failures* result from an object or event having some attribute that contradicts some restriction on its values. Three omission errors have also been identified: *Belated prediction* occurs after the fact. Some prediction that should have

occurred did not, but only in hindsight is this observation made. *Retrieval failures* occur when a reasoner cannot remember an appropriate piece of knowledge; in essence it represents forgetting. *Input failure* is error due to lack of some input information. To construct the three core types highlighted in this paper (Erroneous Association, Novel Situation, and Incorrect Domain Knowledge), representations for expectation failure, retrieval failure, and incorporation failure are needed. The following subsections provide such representations.

### **Successful Prediction**

An illustration of a simple base type representations is contained in figure 10.<sup>19</sup> Let node A be an actual occurrence of an event, an explanation, or an arbitrary proposition. The node A results from either a mental calculation or an input concept. Let node E be the expected occurrence. The expected node E *mentally-results* from some reasoning trace enabled by some goal, G. Now if the two propositions are identical, so that  $A \supseteq E$ , then a successful prediction has occurred.<sup>20</sup> Though successful prediction produces no learning, there must be a representation for it.

For pedagogical reasons figure 11 oversimplifies much of the representation in figure 10; however it illustrates a successful prediction taken from the example program run. In concrete terms, it says that the arrest scene in the drug bust was a part of an overall interdiction case remembered from experience with smuggling acts by terrorists and that such experience produced the inference that the contraband must have been detected before the arrest. The prediction was later confirmed by the story's final sentence in which the dog detected the drugs.

---

19. Attributes and relations are represented explicitly in these figures. For instance, the ACTOR attribute of an event X with some value Y is equivalent to the relation ACTOR having domain X and co-domain Y.

20. See Cox & Ram (1991) for a summary of interpretation for  $A \subset E$ .

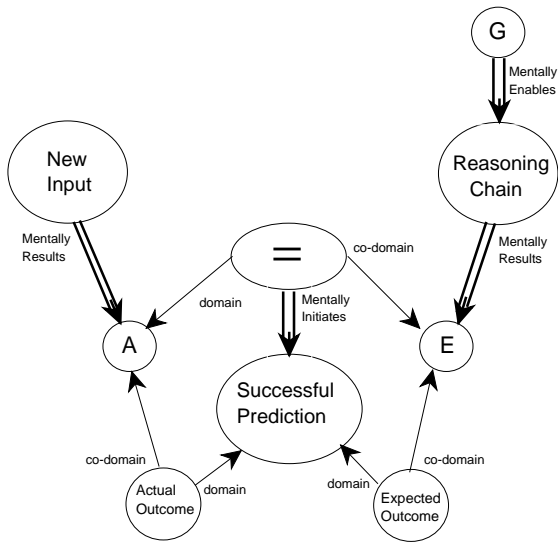


Figure 10: Successful Prediction

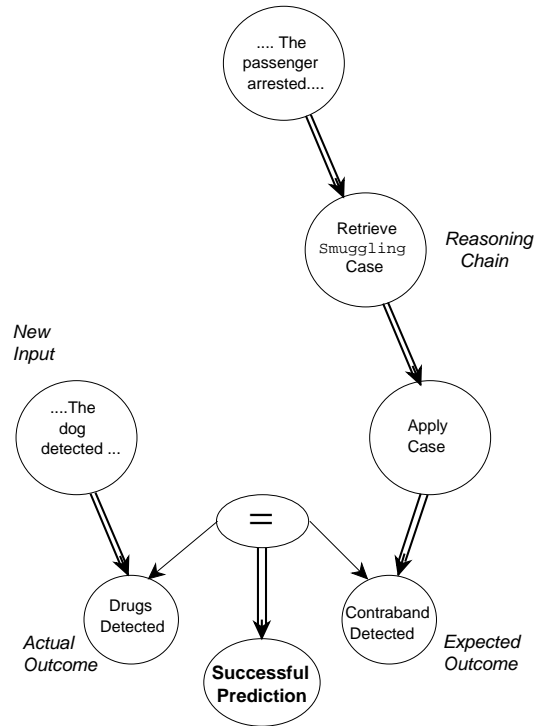


Figure 11: Instantiated Successful Prediction

### Inferential Expectation Failure

Failures occur when  $A \neq E$ . This state exists when A and E are disjoint, or when conflicting assertions within the two nodes conflict. For example, A and E may represent persons, but E contains a relation specifying gender = male, whereas A contains the relation gender = female. Inferential expectation failures (figure 12) occur when the reasoner predicts one event or feature, but another occurs instead. The awareness of expectation failure is initiated by a `not-equals` relation between A and E.

Another simplified example (shown in figure 13) captures the reasoning that was initiated by the sentence specifying the dog barked at the luggage. An abstract explanation concerning threats is therefore retrieved and applied to the input. As a result, the system expects that the dog was prob-

ably threatened. The story later contradicts this prediction, and thus the reasoner experiences a failure.

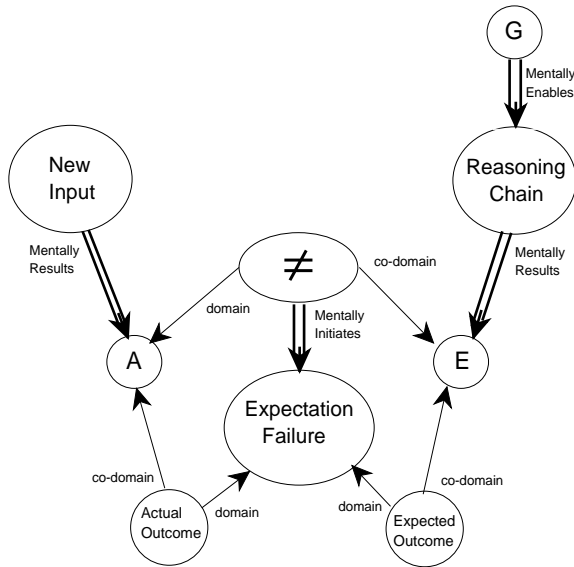


Figure 12: Expectation Failure

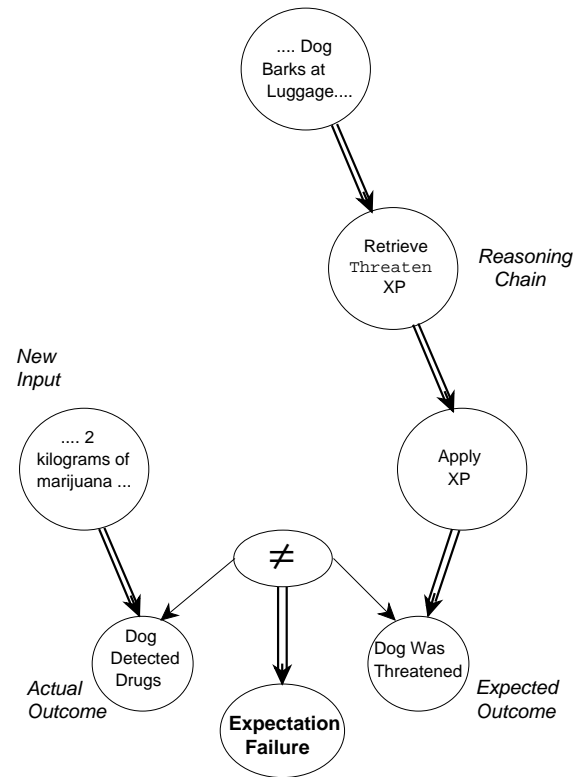


Figure 13: Instantiated Expectation Failure

### Retrieval Failure

As opposed to the representation of expectation failure, in retrieval failure the expectation (E) is absent due to the inability of the system to retrieve any knowledge structure that could produce E (see figure 14). To represent these conditions, Meta-AQUA uses non-monotonic logic values of *in* (in the current set of beliefs) and *out* (out of the current set of beliefs) (Doyle, 1979). Extended values include *hypothesized-in* (weakly assumed in) and *hypothesized (unknown)*. Thus, absolute retrieval failure is represented by  $A (\text{truth} = \text{in}) = E (\text{truth} = \text{out})$ . The relation that identifies the truth value of E as being out of the current set of beliefs *mentally-initiates* the assertion that a retrieval failure exists. Cuts across links in the figure

signify causal relations for which the truth slot of the link is also out.

Figure 15 represents the failure as the understander not being able to retrieve anything that produces the prediction that the dog detects drugs. Knowing that nothing was remembered initiates the reasoner's conclusion that a retrieval failure has occurred.

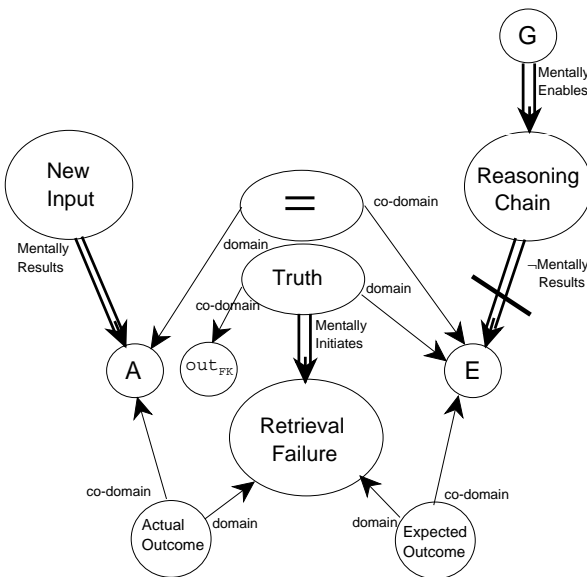


Figure 14: Retrieval Failure

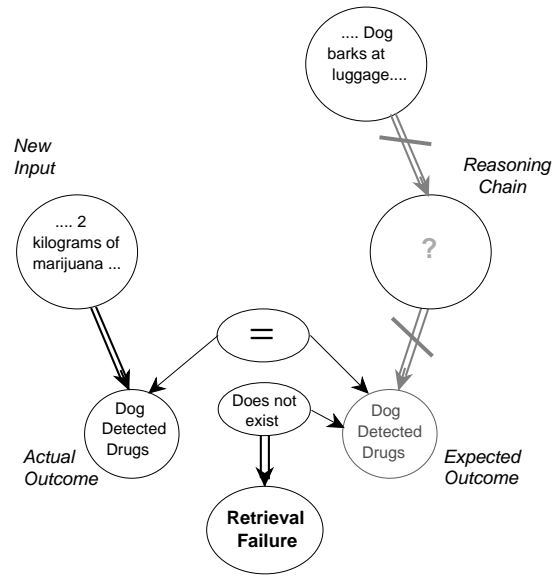


Figure 15: Instantiated Retrieval Failure

### Incorporation Failure

When the incorporation of some input into memory fails due to conflict with the BK, an incorporation failure exists. The conflict produces a not-equals relation between the actual occurrence and a conceptual constraint. This relation mentally-initiates the anomaly (figure 16). Such anomalies are used to identify questions to drive the reasoning and learning processes.

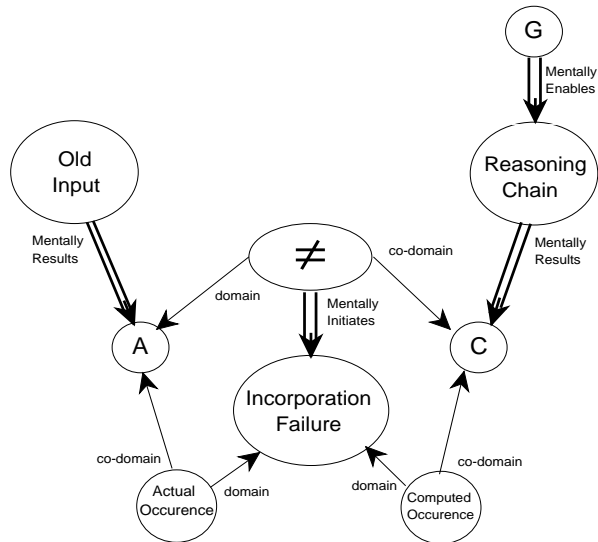


Figure 16: Incorporation Failure

### 4.3.2 Core Class IMXPs

#### Erroneous Association

An *Erroneous Association*, represented by inferential expectation failure, occurs when  $A \neq E$ . This failure occurs when an index has associated some context with part of the BK that produced incorrect inferences. A knowledge organization goal is spawned to adjust the index so that it will still retrieve those structures in the BK when appropriate, but not in future instances similar to the current situation. Learning plans are associated with such goals to execute a specialization algorithm producing a more discriminating index. Because the goal has links to a declarative representation of the reasoning which produced it, the algorithm has access to the context of the error.

#### Missing Association

A *Missing Association* is represented by retrieval failure. Here, an appropriate knowledge structure was not retrieved because there was no index to associate the context with the structure. Thus

some memory element, M, in the BK must be in. The goal associated with the IMXP is to find M. If this can be verified,<sup>21</sup> then the plan that found the structure directs an indexing algorithm to examine the indexes of M, to look for an index compatible with the index calculated for A. If found, this index is generalized so that the current cues provided by the context of A will retrieve M and produce the expectation E. If no such index is found, a new index is computed. If M cannot be found, a reasoning question is raised concerning the possibility that M exists. The question is represented as a knowledge goal and indexed by the context of A, and the process is suspended. If the question ("What was I trying to remember?" or equivalently, "Does M exist and, if so, what is the content of M?") is resumed because of a reminding, then an index can be generated to associate the context, C, with the item M.

In general, core IMXPs formed about a retrieval failure represent various cases of forgetting. As an illustration of the representation for such core IMXPs see figure 17. Different failure types can be represented depending on the content and truth values of the nodes A, E, G, C, I, and M. These types include Novel Situation, Missing Association, Forgotten Goal, and Missing Goal (see table 1). For instance if M, the item to be retrieved, is not actually present in the BK, then the structure signals a Novel Situation. Alternatively, if the item is present, but the index, I, is not present, then a Missing Association, or forgetting, in the normal sense of the word, is represented. Also, if the item to be retrieved was actually a suspended goal, then a Forgotten Goal failure type has occurred. Finally, if G does not exist, then the agent did not try to remember (does not have the goal to retrieve). This instance is called a Missing Goal.

---

21. The presence of M can be determined if the reasoner is reminded of M after a failure. For example, in the case of the stranded motorist she is reminded of the forgotten goal to fill up with gas as the car grinds to a halt. That is, the cues present in the context after the car stops are sufficient to retrieve the goal, whereas those cues present at the time the agent was at the gas station were not. Alternatively an agent can elaborate the current context, C, generating additional cues from which to retrieve the item, M. If this can be performed, then it concludes that it previously forgot the item (the explanation is indeed one of Missing Association), otherwise the Core IMXP represents a hypothesis ("I might have forgotten the item").



is not in the set of beliefs with respect to the FK (i.e., is not in working memory) initiates the state of believing a retrieval failure exists. Some later input may then produce an item A that reminds the agent of the forgotten item. The IMXP indicates that the missing node, I, is probably responsible for the error. The IMXP also suggests that a knowledge acquisition goal should be spawned to produce the correct index. Once a goal scheduler chooses this learning goal, it can select a learning algorithm, such as an index learning algorithm, to either generalize an overly specific index for M, if the index exists at all, or to create one if it does not.

### **Novel Situation**

As discussed in the previous subsection, *Novel Situation* is structurally like a Missing Association, except the node, M, (and thus its associated index) has a truth value of `out`. That is, no item in memory can be retrieved and reasoned with to produce the expectation of a concept like A.

A Novel Situation occurs when M is missing ( $\text{truth} = \text{out}_{\text{BK}}$ ) and E's truth slot is either `hypothesized-in` or `out`. When Meta-AQUA identifies a novel situation it posts a goal to learn a new explanation of the event. The associated plan is to perform EBG on node A, so that the knowledge can be applied to a wider set of future events. The plan also directs an indexing algorithm to the same node so that the new explanation will be retrieved in similar situations.

### **Incorrect Domain Knowledge**

Only one instance of the failure type *Incorrect Domain Knowledge* is currently represented. This failure is an inconsistency between a known fact and a constraint in the BK. Such failures invoke a knowledge acquisition goal to adjust the constraint in the BK. An associated learning plan then tests whether the two assertions (the fact and the constraint) are conceptual siblings. If this is so, then the program will perform abstraction<sup>24</sup> on the constraint, raising it to its parent on the basis of induction. The constraint is then marked as being `hypothesized-in`. The reasoning chain

that led to this hypothesis is indexed off the hypothesis so that the reasoning chain can be retrieved when the constraint is used in future stories. The hypothesis is verified if the anomalous assertion is re-encountered in later situations.

### 4.3.3 Composite Class IMXPs

Composite IMXPs are the abstract structures instantiated in particular instances of reasoning failure. They are used by Meta-AQUA to perform blame assignment and to choose a learning strategy. They represent typical patterns of failure, given the type of reasoner described in this paper. One of the intuitions investigated by this research is the claim that a limited number of major patterns of failure exists, that not just anything will happen, assuming a regular world and a rational reasoner. Thus, the matching necessary to determine what failure pattern corresponds to a given failure event is tractable, whereas the search to construct an analysis of the same failure from first principles is exponential. The power of our approach then comes from the coverage of those situations enumerated in table 1, the quality allowed in our language for expressing them, and the constraints imposed by which combinations are allowed.

In the drug bust example the XP application procedure produces the explanation in figure 18. Note that the reasoning which produced the failure is captured in a simple linear TMXP partially shown in the figure. The Pose Question and XP Application nodes constitute part of a Generate Explanation phase. The TMXP is terminated by the Verify Hypothesis node that results in the expectation failure. This structure was input to the blame assignment phase of learning. An abstract composite IMXP called NOVEL-SITUATION-ALTERNATIVE-REFUTED was bound with this trace to produce the token graphed in figure 18. This entire explanation constitutes the output of blame assignment.

---

24. The use of the term "abstraction" is as defined by Michalski (1991), and can be opposed to that of "generalization." The former is an operation on the `co-domain` of a relation, whereas the latter is an operation on the `domain` of a relation.

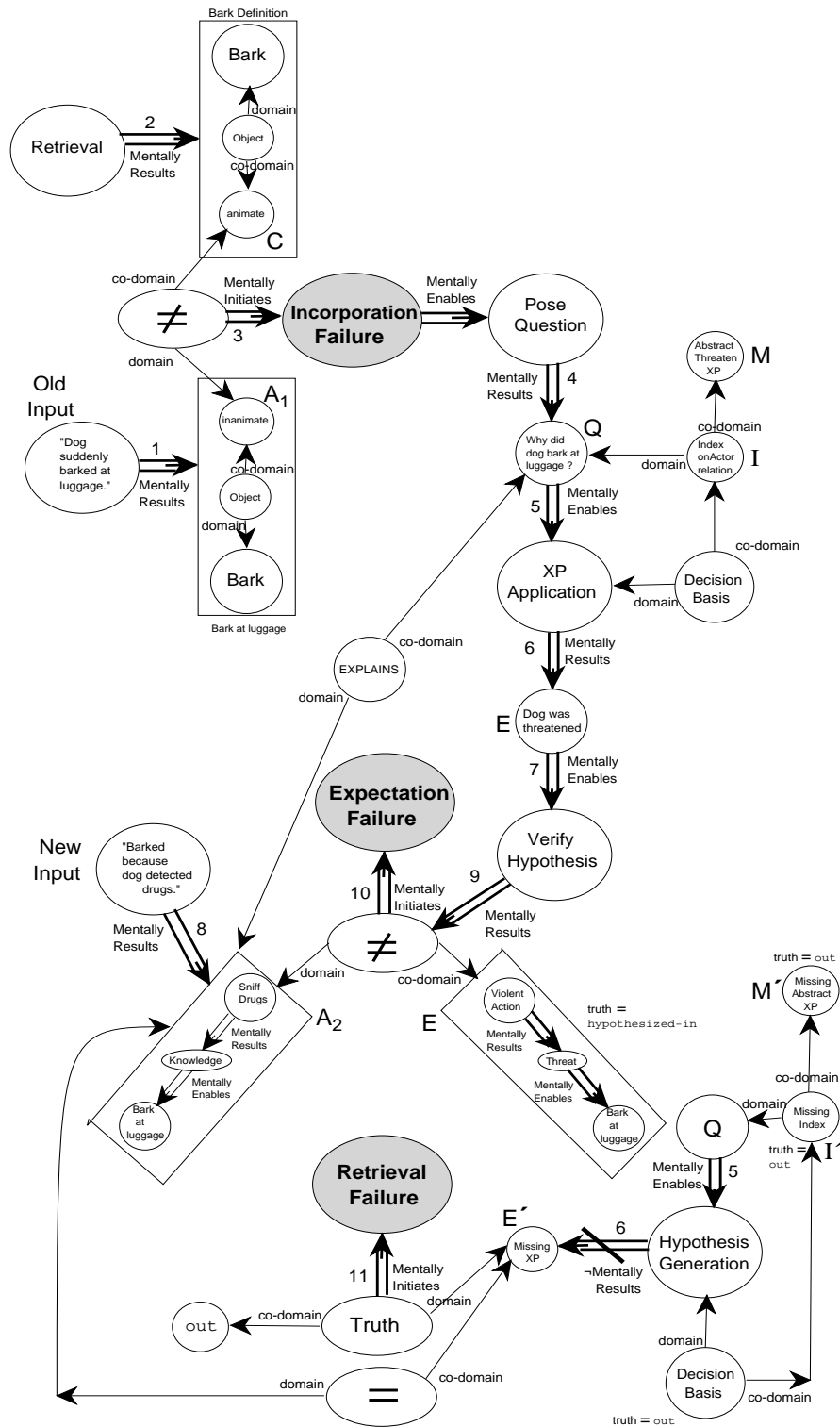


Figure 18: Instantiated Composite Type

The composite Meta-XP consists of three core Meta-XPs: `XP-Novel-Situation` (centered about “Retrieval Failure”), `XP-Erroneous-Association` (centered about “Expectation Failure”) and `XP-Incorrect-Domain-Knowledge` (centered about “Incorporation Failure”). In plain terms the composite says that the dog’s barking at the luggage caused a failure to incorporate the input, because the system had never experienced a dog barking at inanimate objects. This failure resulted in an anomaly that Meta-AQUA tried to explain. The program wondered why the dog barked (posed the question Q) and explained it by producing the hypothesis that the dog was threatened (expectation E). This expectation failed because the later input ( $A_2$ ) was not agreeable with the expectation, constituting an explanation of what happened. What should have happened is indicated in the lower third of the graph. The agent should have applied the detection explanation ( $M'$ ) to produce an alternative explanation ( $E'$ ) which would have agreed with the actual event. It could not, because the knowledge of this type of cause for dogs barking was novel, and hence unknown (out of the set of beliefs with respect to the BK).

The plan, seeking to achieve the knowledge goal spawned by the `XP-Novel-Situation` directs an EBG algorithm to be applied to the explanation of the bark (node  $A_2$ ). Since the detection scene of the drug-bust case and the node representing the sniffing are unified due to the explanation given in S4 (see page 33), the explanation is generalized to drug busts in general and installed at the location of node  $M'$ . The explanation is then indexed in memory, creating a new index ( $I'$ ) associating the explanation concerning detection of contraband with the barking by dogs at containers. The plan for the goal of the `XP-Erroneous-Association` directs an indexing algorithm to the defensive barking explanation (node E). It recommends that the explanation be re-indexed so that it is not retrieved in similar situations in the future. Thus, the index for this XP (node I) is specialized so that retrieval of the threat explanation occurs only when there exists barking by dogs at animate objects, not physical objects in general. The plan achieving the goal of the `XP-Incor-`

rect-Domain-Knowledge directs the system to examine the source of the story's anomaly. The solution is to alter the conceptual representation of "bark" so that the constraint (node C) on the object of dog-barking instantiations is abstracted from animate objects to physical objects.

The abstract composite IMXP from which this explanation is built represents a class of reasoning errors that we claim are quite common. Some overly restricted concept leads a reasoner to think that some related but slightly different concept is unusual. This leads the reasoner to explain the oddity. Because the input is actually an exception to the conceptual definition that the agent has never experienced before, the agent does not have the correct knowledge with which to interpret the input. Instead it chooses some related but causally misleading explanation with which to understand the input. For example, the AQUA program believes that those who commit suicides must be depressed. Thus, when it reads the story of Hamida an anomalous event drives it to explain the action. If the system attempts to explain the suicide as a fanatical event, but later finds out that Hamida was actually blackmailed into the deed, then a problem similar to the drug-bust failure exists. The problem stems from the interaction between an overly restrictive interpretation of suicide, an inappropriate use of the religious-fanatic explanation, and an unknown explanation about suicide bombers. Although AQUA is able to learn about the new explanation in this case, the learning performed is hard-wired into its functions and its flow of control. There is no actual representation of how AQUA processes a story. It therefore can neither understand or make decisions based upon such reasons, nor explain decisions to itself or others. The changes that would be necessary for AQUA to learn from the drug-bust example would involve domain representations mainly, but to learn from another pattern of failure would require major changes to the structure of the AQUA program itself. Contrastingly, Meta-AQUA need only add another composite IMXP and the requisite learning algorithms if not already present.

## **5 CONCLUSIONS**

In summary there are a number of expected contributions from this research. As outlined by this text, the final goal is to establish a specific solution to the strategy selection problem. A more general contribution is to produce a computational model of introspection, and furthermore, to establish the conditions under which such an approach to reasoning and learning is productive. In order to build such a model, a language of mental events and mental causality must be constructed. Thus, a content theory and process theory of both question-driven story understanding and learning must be fully developed.

The remainder of this section will compare the models of understanding and learning as described in the previous sections. Related research in both the artificial intelligence and the psychological communities is then outlined. Appendix A will close the proposal with a discussion of evaluation measures to test the theory and a tentative plan with which to execute the research.

### **5.1 Comparison of Learning and Understanding**

Throughout this exposition numerous parallels have been drawn between introspective learning and understanding. Compare figures 5 and 8, for example, which show the phases of learning and understanding as modeled in our work. This paper has argued, given a multistrategy approach, that a good strategy for both is to identify anomalies, then generate some response to the anomaly, then test the response. The augmented generate-and-test paradigm fits equally well. Both are concerned with selecting a strategy, rather than applying a particular one. Both models are highly top-down and goal-driven. As many of the arguments advanced in this paper show, goals are essential for both focus and direction.

Both the form and the function of the generation phases in learning and understanding are similar (see figure 19). The structure of both is to take some unusual input (reasoning failure or incongruous concept), elaborate the input, generate some goal that provides focus for the process, then change some knowledge base to achieve the function of the process.

There are a number of differences, of course, between learning and understanding. For example, understanding can be likened to recovery; learning can be likened to repair. In the planning literature a number of researchers have made the distinction between recovery and repair (see, for example, Owens, 1991; Hammond, 1989). When a plan fails, the planner must recover from the error so additional progress can be made toward the goal. After recovery, the plan needs to be repaired and stored again in memory, so that the plan failure will not recur. For example (taken from Owens, 1991), if an autonomous robot vehicle finds an expected fuel cache missing and thereby runs out of gasoline, it must first recover from the potentially threatening situation by obtaining fuel. Owens claims that the explanation of the failure will dictate the means of recovery. Therefore, if the robot concludes that it cannot find the gasoline because it is lost, then it should recover by obtaining orientation information; whereas if it explains the fuel's absence because of theft, then the recovery taken will involve turning back or calling for assistance. The repair (to adjust its plans and the information upon which the plan was based) also follows from the explanation of the failure. For instance, if the robot previously considered taking on extra fuel, but did not, because it assumed that the fuel cache would be at the proper location and easy to find, then this explanation of its decision would lead the system to modify its knowledge concerning the persistence of fuel caches. This modification would bias it toward conservative decisions in the future, and thus less likely to repeat the failure.

This difference between recovery and repair, can be applied to the processes of understanding and learning in a complementary manner. The understanding process requires a recovery phase when

it fails. If some explanation does not work, then first there is a need to create a new explanation or somehow to seek one out. Once the correct (or more useful) explanation has been derived, the system needs to learn from the experience by repairing its knowledge, so as not to repeat the failure. Thus, as seen in figure 19, the understanding process operates on the FK to instill the change that removes the anomaly (thus constituting the recovery); whereas the learning process operates on the BK, producing a repaired knowledge base with which the failure will not be as likely in future similar situations. The recovery is a system's response to anomalous input from the outside world that its knowledge could not adequately understand, whereas the learning is a response to the mental world's inadequacy.

There are good functional reasons for having an explicit input analysis stage in both learning and understanding. Most programs accept cleanly defined problems as input, where there exist little ambiguity and sharp distinctions concerning what needs to be done. In more realistic systems there is a need for problem elaboration to clarify what may actually be ill-defined tasks. For example, the task of design is usually well defined relative to many understanding and learning tasks. A designer is often given very specific design specifications from which an artifact must be assembled or created. The task of coming up with such specifications is not usually part of the design system. Specifications are usually provided by the user or programmer. In comprehension tasks such as story understanding, however, the problems are not usually so well defined. In learning, the problem of recovery is to modify the story representation in such a way that the anomaly is coherent with respect to the rest of the story and the system's BK. This specification is so broad that either the programmer must be very clever, so as to include the specifications implicitly, or the explanation must be somewhat trivial. To narrow the range of behaviors appropriate for recovery, then, is to elaborate the input anomaly, so as to identify what went wrong and why.

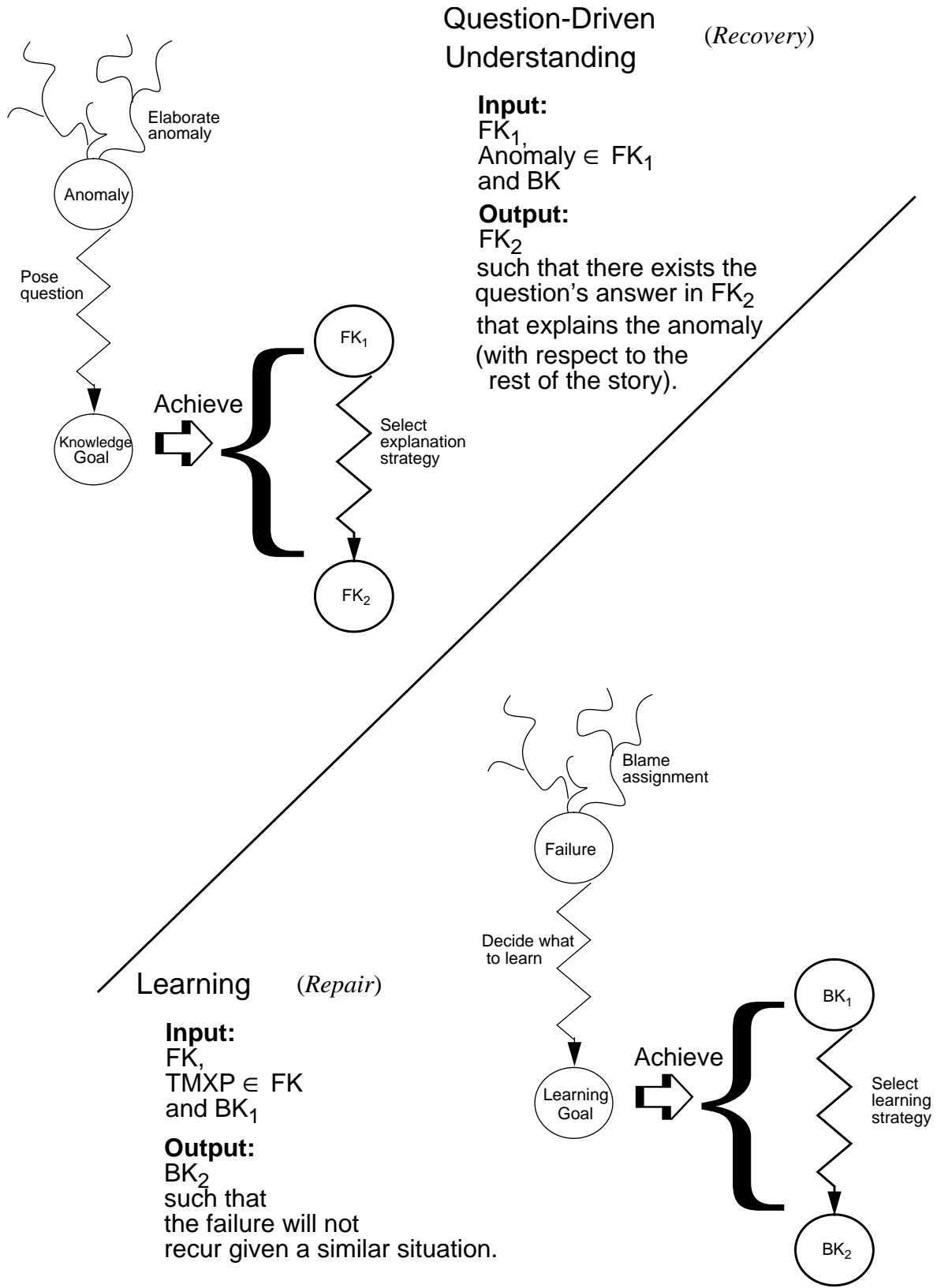


Figure 19: The Parallels in Learning and Understanding

## 5.2 Artificial Intelligence Related Research

Our approach to using an analysis of reasoning failures to determine what needs to be learned is similar to Mooney and Ourston's (1991) EITHER system, Park and Wilkins' (1990) MINERVA program, the CASTLE system of Birnbaum et al. (1990), and Stroulia and Goel's (1992) META-ROUTER, but with some important differences. We have focused on the use of meta-models for explicit representation of domain knowledge and of reasoning processes in an integrated, multi-strategy reasoning and learning system. Unlike Mooney and Ourston and Park and Wilkins, we have not assumed a single reasoning paradigm (logic-based deduction and rule-based expert systems, respectively) in terms of which failure situations and learning strategies are characterized. Rather, it is the architecture that provides a basis for a higher-level characterization, which in turn could be implemented in different ways depending on the reasoning paradigm. In fact, Meta-AQUA uses various reasoning methods, primarily case-based reasoning. Birnbaum et al. (1990) focus on the process of blame assignment by backing up through justification structures, but do not emphasize the declarative representation of failure types. They explicitly model, however, the planner itself. They also explicitly model and reason about the intentions of an actor in order to find and repair the faults that underlie a planning failure (see Freed et al., 1992). Though much is shared between CASTLE and Meta-AQUA in terms of blame assignment (and to a great extent CASTLE is also concerned with deciding what to learn), they do not use failure characterizations to formulate explicit learning goals or to select learning strategies in a multistrategy learning system.

Owens (1991) has enumerated a failure taxonomy of planning errors that is largely subsumed by the failure taxonomy of this research, with one prominent exception. Owens' taxonomy includes failures attributable to time constraints, such as plan failures due to insufficient time to complete the task and achieve the goal. Yet it is an open question how the framework presented here can represent temporal failure types. Collins et al. (1992) have modeled the time constraints of a planner

in a chess domain, however, the use of a logic formalism and the cognitively implausible construct of priority queues makes direct transfer to Meta-XP theory unlikely.

Finally, our work focuses on reasoning failures, and not only on performance failures (in both the Birnbaum et al. and Owens' cases, planning failures). Stroulia and Goel's approach focuses on a design stance characterization of the reasoner as a device, whereas our approach, as with the approach of Birnbaum et al. take a more intentional stance toward the reasoner.<sup>25</sup> The analysis of Stroulia (1992), characterizing the ways in which such a device could fail, yields a taxonomy of failure types similar to ours. However, like the previous studies, they too do not use declarative characterizations of reasoning failures to formulate explicit learning goals. Despite these differences, we emphasize that the above approaches have much in common with Meta-AQUA.

### 5.3 Psychological Influences and Support

*"One form of metacognition - metacomprehension - addresses the abilities of individuals to adjust their cognitive activity in order to promote more effective comprehension. We have been interested in a specific aspect of metacomprehension - namely, the manner in which questions generated by sources external to the learner (i.e., from the teacher or text), as well as those questions generated by the learners themselves, serve to promote their comprehension of text."*

--Gavelek & Raphael (1985).

The literature on metacognition (cognition about cognition where the self is a referent) provides a wide array of influences and support that bear on the research presented in this paper. Our model of integrated introspective learning makes several claims about the nature of learning, reasoning, and introspection that are supported by research in psychology and metacognition. As suggested by the quotation above, there is a special relation between metacognition, question asking and text

---

25. Interestingly, Collins et al. argue from both a design stance and an intentional stance.

understanding. In effect, human learners use question-asking and question-answering strategies to provide an index into their level of comprehension of a given piece of text. This metacognitive feedback helps readers find areas where their understanding of the story is deficient, and thus where greater processing is necessary. Such a perspective supports our claim that question generation is a key activity in text comprehension, and also that meta-level processing is important in such a learning context.

The Meta-Explanations in this approach are similar to self-explanations (Chi & Van Lehn, 1991; Pirolli & Bielaczyc, 1989; Van Lehn et al., 1992). This research shows that formulation of self-explanations while understanding input examples significantly improves the subjects' ability to learn from the examples. One difference between the two approaches is that self-explanations are self-generated explanations about the world, whereas meta-explanations are explanations about the self. Despite the differences, experimental results in the metacognition literature suggest that introspective reasoning of the kind proposed here can facilitate reasoning and learning.<sup>26</sup>

Wellman (1983, 1985) views human metacognition not as a unitary phenomenon, but rather as a multifaceted theory of mind. Metacognition involves several separate but related cognitive processes and knowledge structures that share as a common theme the self as referent. Such a theory of mind emerges from of an awareness of the differences between internal and external worlds, from the perception that there exist both mental states and events that are quite discriminable from external states and events, but that mental states have a close tie to the outside world. This theory encompasses a number of knowledge classes considered by Wellman to be psychological variables: *person variables* that deal with the individual and others (for example, cognitive psychologists can recall many facts about cognition, whereas most people cannot), *task variables*, which

---

26. See Cox (1993) for a review and critique of the psychological research on metacognition in the area of problem solving and the relevance of Meta-XP theory to such research.

concern the type of mental activity (for example, it is more difficult to remember nonsense words than familiar words), and *strategy variables* that relate to alternative approaches to a mental task (e.g., to remember a list it helps to rehearse). Finally, Wellman's theory includes a self-monitoring component, whereby people evaluate their levels of comprehension and mental performance with respect to the theory and the norms the theory predicts.

There are four important ways that such metacognitive knowledge and capabilities bear on work in introspective learning. First, and foremost, is the emphasis on cognitive self-monitoring. This behavior is the human ability to read one's own mental states during cognitive processing (Flavell & Wellman, 1977; Wellman, 1983, 1985). A person has a moment-by-moment understanding of the content of one's own mind, and therefore, both internal cognitive feedback and a judgement of progress (or the lack thereof). Garner (1987) has argued that metacognition and comprehension monitoring are important factors in the understanding of written text. Reading comprehension is therefore considered to be chiefly an interaction between a reader's expectations and the textual information. Psychological studies have also confirmed a positive correlation between meta-memory and memory performance in cognitive monitoring situations (Schneider, 1985; Wellman, 1983). This evidence directly supports the conviction that there must be a second-order introspective process that reflects to some degree on the performance element in an intelligent system, especially a system involved in understanding tasks.

Second, our Meta-XP theory places a heavy emphasis on explicit representation. Trains of thought, as well as the products of thought, are represented as metaknowledge structures, and computation is not simply the calculated results from implicit side-effects of processing. This emphasis is echoed in Chi's (1987) argument, that to understand knowledge organization and to examine research issues there must be some representational framework. Although diverging from the framework suggested by Chi, Meta-XP theory provides a robust form with which to represent knowledge

about knowledge and knowledge about process. For example, Meta-XPs can represent the difference between remembering and forgetting (Cox & Ram, 1992b). Since forgetting is characterized by the absence of a successful outcome, it is quite difficult to capture in most representational languages. In general, forgetting is a neglected issue in AI and computational learning research, yet forgetting is a significant issue in the metamemory literature (Spear, 1978; Wellman & Johnson, 1979). This paper has argued for its importance, however, and specified a formalism that captures such memory and processing phenomena.

Third, because the approach taken by the introspective learning paradigm clearly addresses the issues of memory organization and the indexing problem, it can assign blame to errors that occur from mis-indexed knowledge structures and poorly organized memory. As Ram and Cox (to appear) argue, the memory organization of suspended goals, BK, and reasoning strategies are as important in determining the cause of a reasoning failure as are the goals, propositions and strategies themselves. Thus, memory retrieval and storage are relevant in deciding what to learn and which learning strategy is appropriate. This claim is supported by the metamemory community's focus on organizational features of memory and their relation to the human ability to know what one knows, even in the face of an unsuccessful memory retrieval.

Finally, both metacognition theory and Meta-XP theory address the issue concerning a person's ability to assess the veracity of their own responses. In addition, because a person has a feeling of knowing, even when recall is blocked, the agent can make efficient use of search. Thus, search and elaboration is pursued when an item is on the "tip of the tongue" and abandoned when an item is judged unfamiliar. This search heuristic provides efficient control of memory and avoids the combinatoric explosion of inferences (Lachman, Lachman & Thronesbery, 1979). Modelling this dimension of human metaknowledge requires a two-layered memory architecture. In the lower layer would be the actual memories, cases and propositions. The second layer would be a

metamemory, which stores memories of the first layer. This model would account for the human ability to know what one knows. This second layer might be composed of Trace Meta-XPs, perhaps representing the memories of retrieving a past memory. Much work remains to be done to implement and test such a model, but our framework allows a natural integration of these ideas into the existing implementations.

One of the major differences between the manner in which humans learn and the manner in which machines do is that humans perform dynamic metacognitive monitoring or self-evaluation. Humans often know when they are making progress in problem solving, even if they are far from a solution, and they know when they have sufficiently learned something with respect to some goal (Weinert, 1987). They know how to allocate mental resources and can judge when learning is over. Many reviews (e.g., Chi, 1987; Schneider, 1985; Wellman, 1983) cite evidence for such claims. Research in Meta-XP theory is a step in the direction of adding this metacognitive monitoring capability to AI systems.<sup>27</sup>

---

27. It should be noted that the learning strategies selected by programs such as Meta-AQUA are at a finer level of granularity than those examined by much of psychology. For example, it would be misleading to assert that the types of learning strategies studied by the metacognition community are similar to index learning, explanation-based generalization, and other learning strategies in Meta-AQUA. Instead, metacognition research focuses on a person's choice of strategies at the level of elaboration or rehearsal. However, many of the results from metacognition research do support the overall approach taken in this thesis. Although our research is currently building computer systems at what might be called the micro-level, eventually, it would be desirable to build systems that integrate the kinds of behavior exhibited by human learners at the macro-level as well.

## **A APPENDIX: Research Agenda**

This appendix provides a brief sketch of how Meta-XP theory might be evaluated both quantitatively and qualitatively, both computationally and psychologically, and how the theory can be further implemented.

### **A.1 Evaluation**

Part of the research will be to develop good evaluation criteria. Unlike other domains in machine learning, such as classification tasks and problem solving tasks where optimal or provably correct solutions to well defined tasks exist, it is unclear how to evaluate understanding and comprehension tasks. The question of what it means for an agent to understand something is not easily answerable in general terms. Comprehension problems are different from more quantitative problems such as mathematics or puzzle solving. There is a definitive answer to the question, "What is  $2 + 2$ ?"<sup>28</sup>, but to truly answer the question, "Why did the actor commit suicide?" requires interpretation and inference with respect to the world, and perhaps an understanding of why the question is being asked. There simply is no one correct reason, rather there are many answers at different levels of specification, depending upon the context and the knowledge of the reasoner.

The first and last sentences of the abstract form the testable hypothesis of this research:

The thesis of this proposal is that introspection facilitates learning by providing a basis for identifying what needs to be learned and for selecting an appropriate learning algorithm. ... Thus, the object of the proposed research is to develop both a content theory and a process theory of introspective multistrategy learning and to establish the conditions under which such an approach is fruitful.

---

28. Of course, this statement assumes that the addition is not being conducted in base 4 or less.

The claims are therefore that

1. Introspection facilitates learning.
2. There are conditions for which claim #1 is true, and other conditions for which it is not true.
3. The process and content theories constitute a reasonable model of introspection.

Although much work remains in determining the best evaluation for these claims, a beginning is as follows: Holding other factors constant, if the rate of improvement in a performance task can be shown to be greater with introspection than without, introspection must be responsible for the improved performance. Second, if the distributions of the kinds of failures generated by the performance task change the nature of the differences in the learning curves in the previous measure, then the applicability conditions can be established for indicating when there is a gain for introspection and when there is not. Finally, if, with minimal changes, the Meta-AQUA model can cover real human data on metacognition, then the theory is reasonable.

### **A.1.1 Computational Empirical Evaluation**

One promising approach to establishing the thesis is to perform ablation studies: to show different learning function with and without introspection. The model of introspection presented here relies on the four steps in learning provided by IMXPs, which are to perform input elaboration or blame assignment, to post a learning goal, to select a strategy, and then to perform the strategy. A more reflexive, or automatic, way to choose the learning strategy is to simply identify a failure, then select a learning strategy based on a static association between fault and method. Thus, given a number of input stories, learning curves can be established as a function of the number of inputs.

Two performance measures are obvious candidates for this evaluation. The number of concepts or indexes the system acquires, modifies or deletes provides one empirical measurement. The number

of anomalies generated in stories reflects how well it understands a given domain. The thesis that introspection facilitates learning can then be established, if the curve of introspective strategy selection indicates an improvement over the curve of reflexive strategy selection. This independent variable (use of introspection) is then manipulated by removing all IMXPs from memory. In their stead direct associations of failure and strategy replace the sequence of input elaboration (blame assignment), followed by the posting of learning goals and then strategy selection.

Pazzani (1991) has established a relatively unbiased approach to generating test data. A program called Tale-Spin is used to generate pseudo-random stories in some domain, by providing the program with a given domain representation. This method is an improvement over hand-coded input, since it reduces the bias introduced by the programmer. Given various input stories, Meta-AQUA's task is to understand the stories and to acquire the domain knowledge of Tale-Spin. Meta-AQUA is given a similar domain theory possessing gaps, incorrect assertions, and missing or incorrect indexes. The expectation is that the number of items to be learned can be quantified as a difference between Tale-Spin's knowledge and Meta-AQUA's knowledge. So, as Meta-AQUA's domain theory approaches Tale-Spin's, the number of anomalous inputs should be reduced as a function of the number of inputs processed. This process will produce a learning curve. The input can then be repeated on the program with introspection disabled.

### **A.1.2 Cost-Benefit Analysis**

Research into introspection has long been controversial. Not only has it been claimed by some that learning can be explained without such a theoretical construct, but it has been empirically shown by Wilson and Schooler (1991) and others that there are conditions under which introspection actually degrades performance. Likewise, computational introspection is not expected to be effective under all circumstances. Under extremely complex situations, or in data-impooverished circumstances, deciding on a specific learning goal may be intractable. Identifying these conditions is

therefore a quite desirable goal.

The space of applicability conditions for introspection is expected to emerge from the types of failures possible in Meta-AQUA, as organized by table 1. It has already been shown through the existing implementation that introspection is possible in Meta-AQUA. 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. So an analysis of the interaction of failure types in the table should result in a set of complex failures that can be programmed into Tale-Spin so as to produce various distributions of errors. It is expected that certain failure combinations and distributions will be difficult for Meta-AQUA to learn from. As with the ablation study, measures with and without introspection provide the independent variable for the evaluation of learning. The results should determine those combinations that are either impossible to recover from, or those for which a simpler reflexive or associative approach is more suited.

### **A.1.3 Psychological Empirical Evaluation**

Recker and Pirolli (to appear) have shown that a Soar-based model of learning called SURF can explain individual differences exhibited by human subjects while learning to program in LISP using instructional text. The difference that accounted for much of the variability was self-explanation strategies. Those students who explained problems to themselves during comprehension of the instructions performed well on a subsequent performance task consisting of LISP programming exercises. The students who did not exhibit this behavior were not as likely to excel in the LISP task. The SURF model predicted such differences. The model took into account, however, only domain-related elaborations; whereas subjects exhibited other self-explanations that the model did not cover. In particular, some subjects seemed to exploit metacognitive feedback, like comprehension monitoring, in order to judge when to learn. If self-reflection on the states of a subject's comprehension of the instruction indicated an understanding failure, then this was sometimes used as

a basis to form a learning goal.

These data seem well-suited for implementation in the Meta-AQUA framework, and have the virtue of already existing. An additional virtue is that both the data and the Meta-AQUA model arose independently. If Meta-AQUA can be changed with minimal effort, including the addition of the learning strategies suggested above, then there is evidence that Meta-XP theory is a reasonable model of introspection. In addition, the model gains credibility if it can be extended to a new performance domain: instructional text comprehension in addition to story understanding. If the changes to the model are significant, however, then the believability of our theory as an approximate model of human metacognition is lessened.

Part of this evaluation will include a comparison to the Soar/SURF model of learning. Although the fundamental philosophies and approaches of multistrategy learning and learning via chunking appear to be incongruent, it is anticipated that the differences can be reconciled, and that a complete accounting of learning will likely include both theories. In addition, Soar claims to possess a meta-level architecture (see Rosenbloom, Laird, and Newell, 1988). These claims must also be evaluated with respect to the theory embodied in Meta-AQUA.

## **A.2 Plan**

As described briefly in section 4.1, “Implementation and Example,” and in more detail in Cox & Ram (1991, 1992a) and Ram & Cox (to appear), a significant implementation of Meta-AQUA already exists that performs story understanding, blame assignment, strategy selection, and repair. A significant amount of domain knowledge has been represented in a frame language representational system written by the author. A significant amount of work remains to be accomplished, however, before the dissertation can be written.

The initial step in achieving this thesis is to complete the representations that comprise the taxonomy of reasoning failures in table 1. Many of the cells of the table are currently represented, including some not described in this document. At present, representations exist for the following columns: domain knowledge, knowledge selection, and input. In addition, representations exist for the missing-goal and forgotten-goal cells. It is anticipated that the most difficult representations to generate will be the processing strategy column. To represent a function that is incorrect will probably entail having a representation for process itself, and the recursive behavior that it exhibits. Either the Structure-Behavior-Function language of Stroulia (1992) or the formalism of Kuokka (1990), suitably augmented to accommodate goals, may be useful for the task; any representation that provides sufficient coverage will probably be similar to one of these models.

As briefly mentioned at the end of section 3.4, “Taxonomy of Reasoning Failures,” relations among the cells of table 1 exist. If these relations are ascertained, then they may be used to constrain inference by the system. A trivial constraint is that it cannot be true that a dimension, such as input, is both correct and wrong. Heuristics must also be developed to distinguish syntactically similar failure patterns. For example, a useful heuristic would be a rule that helps to determine whether a failure is due to noisy input or rather a novel situation.

A copy of Pazzani’s implementation of Tale-Spin exists on file at the Georgia Institute of Technology. Many changes, including knowledge representation as well as procedural changes, will be required to implement the problem generator briefly described in appendix subsection A.1.2, “Cost-Benefit Analysis.”

At this time, the planning mechanism and the learning plans that compose the learning strategies associated with particular knowledge goals are treated very simplistically. To be complete, much more work remains to be accomplished. Two possibilities exist with which to implement a robust

planner. Either a case-based planner could be produced (or borrowed and adapted), for example, the CHEF program described by Hammond (1989), or a traditional non-linear planner could be incorporated.

## **ACKNOWLEDGEMENTS**

The National Science Foundation supported this research through grant #IR-9009710. The Georgia Institute of Technology also provided valuable support. My proposal committee gave me both inspiration and guidance that reflects heavily in the content of this document. Sue Farrell proofed two drafts of the proposal, and suggested innumerable changes that improved the presentation by an order of magnitude. Many helpful comments and productive discussions were offered by both Mimi Recker and Eric Domeshek. Constructive criticism during practice presentations was extremely enlightening, especially the feedback from Eleni Stroulia, Justin Peterson, Sam Bhatta, Kavi Mahesh, Anthony Francis, and the IGOR research group.

## REFERENCES

- Anderson, J. R. (1983). *The Architecture of Cognition*. Harvard University Press, Cambridge, MA.
- Bhatta, S., and Ram, A. (1991). Learning Indices for Schema Selection. In M. B. Fishman (ed.), *Proceedings of the Fourth Florida Artificial Intelligence Research Symposium (FLAIRS)*, Cocoa Beach, FL, (April), pp. 226-231.
- Birnbaum, L. (1986). "Integrated Processing in Planning and Understanding," PhD Thesis, Research Report 489, Yale University, New Haven, CT (December).
- Birnbaum, L., and Collins, G. (1984). Opportunistic Planning and Freudian Slips. In *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*. Boulder, CO, pp. 124-127.
- Birnbaum, L., and Collins, G., Freed, M. and Krulwich, B. (1990). Model-Based Diagnosis of Planning Failures. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, MA, pp. 318-323.
- Booker, L. B., Goldberg, D. E. and Holland, J. H. (1989). Classifier Systems and Genetic Algorithms. *Artificial Intelligence*, Vol. 40, pp. 235-282.
- 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: An artificial intelligence approach*, Vol 2. Morgan Kaufmann Publishers, San Mateo, CA, pp. 371-392.
- Carbonell, J. G., Knoblock, C. A., and Minton, S. (1991). PRODIGY: An integrated architecture for planning and learning. In K. Van Lehn (ed.), *Architecture for Intelligence: The twenty-second Carnegie Mellon symposium on cognition*. Lawrence Erlbaum, Associates, Hillsdale, NJ, pp. 241-278.
- Chandrasekaran, B. (1989). Task-Structures, Knowledge Acquisition and Learning, *Machine Learning*, Vol. 4. pp. 339-345.
- Chi, M. T. H. (1987). Representing Knowledge and Metaknowledge: Implications for interpreting metamemory research. in F. E. Weinert and R. H. Kluwe (eds.), *Metacognition, Motivation, and Understanding*. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, pp. 239-266.
- Chi, M. T. H., and Van Lehn, K. A. (1991). The Content of Physics Self-Explanations. *The Journal of the Learning Sciences*, Vol. 1, No. 1, pp. 69-105.
- Clancey, W. J. (1991). The Frame of Reference Problem in the Design of Intelligent Machines. In K. Van Lehn (ed.), *Architecture for Intelligence: The twenty-second Carnegie Mellon symposium on cognition*. Lawrence Erlbaum, Associates, Hillsdale, NJ, pp. 357-423.
- Collins, G., Birnbaum, L., Krulwich, B., and Freed, M. (1992). The Role of Self-Models in Learning to Plan. In A. Meyrowitz (ed.), *Machine Learning: Induction, analogy and discovery*. Kluwer Academic Publishers. (Also available as Technical Report 24, Institute of the learning Sciences, Northwestern University, Evanston, IL, April, 1992).

- Cox, M. T. (1991). Reasoning about Reasoning via META-XPs. Unpublished.
- Cox, M. T. (1992). Toward an Epistemological Treatment of the Blame Assignment Problem. Unpublished.
- Cox, M. T. (1993). Metacognition, Problem Solving and Aging. Submitted to *The Psychology Graduate Student Journal (PSYCGRAD)*.
- Cox, M. T., and Ram, A. (1991). Using Introspective Reasoning to Select Learning Strategies. In R. S. Michalski and G. Tecuci (eds.), *Proceedings of the First International Workshop on Multi-strategy Learning*. Harpers Ferry, WV, (November), pp. 217-230.
- Cox, M. T., and Ram, A. (1992a). Multistrategy Learning with Introspective Meta-Explanations. In D. Sleeman and P. Edwards (eds.), *Machine Learning: Proceedings of the ninth international conference (ML92)*, Aberdeen, Scotland, (July 1-3), pp. 123-128.
- Cox, M. T., and Ram, A. (1992b). An Explicit Representation of Forgetting. In J. W. Brahan and G. E. Lasker (eds.), *Proceedings of the Sixth International Conference on Systems Research, Informatics and Cybernetics*. Vol. 2 (Advances in Artificial Intelligence - Theory and application), Baden-Baden, Germany, (August 17-23), pp. 115-120.
- Davis, R. (1980). Meta-Rules: Reasoning about control. *Artificial Intelligence*, Vol. 15, No. 3, pp. 179-222.
- Davis, R., and Buchanan, B. G. (1977). Meta-Level Knowledge: Overview and applications. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*. Vol. 2, Cambridge, MA, (August 22-25), pp. 920-927.
- DeJong, G., and Mooney, R. (1986). Explanation-Based Learning: An alternative view, *Machine Learning*, Vol. 1, No. 2, pp. 145-176.
- Domeshek, E. A. (1992). "Do the Right Thing: A component theory for indexing stories as social advice," PhD Thesis, Technical Report 26, Institute for the Learning Sciences, Northwestern University, Evanston, IL (May).
- Doyle, J. (1979). A Truth Maintenance System, *Artificial Intelligence*, Vol. 12, pp. 231-272.
- Flavell, J. H., and Wellman, H. M. (1977). Metamemory. In R. V. Kail, Jr., and J. W. Hagen (eds.), *Perspectives on the Development of Memory and Cognition*. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, pp. 3-33.
- Freed, M., Krulwich, B., Birnbaum, L., and Collins, G. (1992). Reasoning about Performance Intentions. In *Proceedings of Fourteenth Annual Conference of the Cognitive Science Society*, Bloomington, IN, (July 29 - August 1), pp. 7-12.
- Garner, R. (1987). *Metacognition and Reading Comprehension*. Ablex Publishing Corporation, Norwood, NJ.
- Gavelek, J. R., and Raphael, T. E. (1985). Metacognition, Instruction, and the Role of Questioning Activities. In D. L. Forrest-Pressley, G. E. MacKinnon, and T. G. Waller (eds.), *Metacognition, Cognition and Human Performance*. Vol. 2 (Instructional Practices), Academic Press, Inc., New York, pp. 103-136.
- Goel, A. K., and Callantine, T. J. (1991). A Control Architecture for Run-Time Method Selection and Integration. In *Proceedings of the AAAI Workshop on Cooperation Among Heterogeneous*

*Intelligent Agents*, Anaheim, CA (July, 15).

Hammond, K. J. (1988). Opportunistic Memory: Storing and recalling suspended goals, In J. L. Kolodner (ed.), *Proceedings of a Workshop on Case-Based Reasoning*. Clearwater Beach, FL, (May 10-13), pp. 154-168.

Hammond, K. J. (1989). *Case-Based Planning: Viewing planning as a memory task*. Vol. 1 of *Perspectives in Artificial Intelligence*. Academic Press, San Diego, CA.

Hayes-Roth, B., and Hayes-Roth F. (1979). A Cognitive Model of Planning, *Cognitive Science*, Vol. 2, pp. 275-310.

Hinrichs, T. R. (1992). *Problem Solving in Open Worlds*. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ.

Hunter, L. E. (1989). "Knowledge Acquisition Planning: Gaining experience through experience," PhD Thesis, Research Report 678, Yale University, New Haven, CT (January).

Hunter, L. E. (1990). Planning to Learn. In *Proceedings of Twelfth Annual Conference of the Cognitive Science Society*, Cambridge, MA, (July 25-28), pp. 261-276.

Jones, R., and Van Lehn, K. (1991). Strategy Shifts without Impasses: A computational model of the sum-to-min transition. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. Chicago, IL, (August 7-10), pp. 358-363.

Kass, A., Leake, D., and Owens, C. (1986). "SWALE: A program that explains," In R. C. Schank, *Explanation Patterns: Understanding mechanically and creatively*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Keller, R. M. (1986). Deciding What to Learn. Technical Report ML-TR-6, Rutgers University, Department of Computer Science.

Kolodner, J. L. (1984). *Retrieval and Organizational Strategies in Conceptual Memory: A computer model*. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ.

Kolodner, J. L. (1987). Capitalizing on Failure through Case-Based Inference. In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, Seattle, WA, pp. 715-726. (Also available as Technical Report GIT-ICS-87/18, College of Computing, Georgia Institute of Technology, Atlanta, GA).

Konolige, K. (1988). Reasoning by Introspection, In P. Maes and D. Nardi (eds.), *Meta-Level Architectures and Reflection*, North Holland, Amsterdam, pp. 61-74.

Krulwich, B. (1991). Determining What to Learn in a Multi-Component Planning System. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. Chicago, IL, (August 7-10), pp. 102-107.

Kuokka, D. R. (1990). "The Deliberative Integration of Planning, Execution, and Learning." PhD thesis, Report CMU-CS-90-135, Carnegie Mellon University, Pittsburgh, PA.

Lachman, J. L., Lachman, R., and Thronesbery, C. (1979). Metamemory Through the Adult Life Span. *Developmental Psychology*, Vol. 15, No. 5, pp. 543-551.

Laird J. E., Rosenbloom, P. S., and Newell, A. (1986). Chunking in SOAR: The Anatomy of a General Learning Mechanism, *Machine Learning*, Vol. 1, pp. 11-46.

- Leake, D., and Ram, A. (to appear). Goal-Driven Learning: Fundamental issues and symposium report. *AI Magazine*. (Also available as Technical Report 85, Cognitive Science Program, Indiana University, Bloomington, IN, 1993).
- Maes, P. (1988). Issues in Computational Reflection. In P. Maes and D. Nardi (eds.), *Meta-Level Architectures and Reflection*. North Holland, Amsterdam, pp. 21-35.
- Michalski, R. S. (1991). Inferential Learning Theory as a Basis for Multistrategy Task-Adaptive Learning, In R. S. Michalski and G. Tecuci (eds.), *Proceedings of the First International Workshop on Multistrategy Learning*. Harpers Ferry, WV, (November), pp. 3-18.
- Michalski, R. S., and Tecuci, G. (eds), (1991). *Proceedings of the First International Workshop on Multistrategy Learning*. Harpers Ferry, WV, George Mason University, (November).
- Michalski, R. S., and Tecuci, G. (eds.) (to appear) *Machine Learning: A multistrategy approach IV*, Morgan Kaufmann, Los Altos, CA.
- Minsky, M. L. (1963). Steps Towards Artificial Intelligence. in E. A. Feigenbaum & J. Feldman (eds.), *Computers and Thought*. McGraw Hill, New York, pp. 406-450.
- Mitchell, T., Utgoff, P. E., and Banerji, R. (1983). Learning by Experimentation: Acquiring and refining problem-solving heuristics. In R. S. Michalski, J. G. Carbonell and T. M. Mitchell (eds.) *Machine Learning: An artificial intelligence approach*. Morgan Kaufmann, Inc., Los Altos, CA, pp. 163-189.
- Mitchell, T., Keller, R., and Kedar-Cabelli, S. (1986). Explanation-Based Generalization: A unifying view, *Machine Learning*, Vol. 1, No. 1, pp. 47-80.
- Mooney, R., and Ourston, D. (1991). A Multistrategy Approach to Theory Refinement. In R. S. Michalski and G. Tecuci (eds.), *Proceedings of the First International Workshop on Multistrategy Learning*. Harpers Ferry, WV, (November), pp. 217-230.
- Newell, A., and Simon, H. A. (1972). *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, NJ.
- Owens, C. (1991). A Functional Taxonomy of Abstract Plan Failures, In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. Chicago, IL, (August 7-10), pp. 167-172.
- Park, Y. T., and Wilkins, D. C. (1990). Establishing the Coherence of an Explanation to Improve Refinement of an Incomplete knowledge Base. In *Proceedings of the Eighth National Conference on Artificial Intelligence*. Boston, MA, pp. 318-323.
- Pazzani, M. (1991). Learning Causal Patterns: Deliberately overgeneralizing to facilitate transfer. In R. S. Michalski and G. Tecuci (eds.), *Proceedings of the First International Workshop on Multistrategy Learning*. Harpers Ferry, WV, (November), pp. 19-33.
- Pirolli, P., and Bielaczyc, K. (1989). Empirical Analyses of Self-Explanation and Transfer in Learning to Program. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*. Ann Arbor, MI, pp. 450-457.
- Punch III, W. F. (1991). TIPS (Task-Integrated Problem Solver), a Task-Specific Integration Architecture for Heterogeneous Agents. In *Proceedings of the AAAI Workshop on Cooperation Among Heterogeneous Intelligent Agents*, Anaheim, CA, (July, 15).

- Ram, A. (1989). "Question-Driven Understanding: An integrated theory of story understanding, memory and learning," PhD Thesis, Research Report 710, Yale University, New Haven, CT (May).
- Ram, A. (1990). Decision Models: A theory of volitional explanation, In *Proceedings of Twelfth Annual Conference of the Cognitive Science Society*, Cambridge, MA, (July 25-28), pp. 198-205.
- Ram, A. (1991). A Theory of Questions and Question Asking. *The Journal of the Learning Sciences*, Vol. 1, Nos. 3&4, pp 273-318.
- Ram, A. (1993). Indexing, Elaboration and Refinement: Incremental Learning of Explanatory Cases. *Machine Learning*. Vol. 10, pp. 201-248.
- Ram, A., and Cox, M. T. (to appear). Introspective Reasoning Using Meta-Explanations for Multistrategy Learning. In *Machine Learning: A multistrategy approach IV*, Michalski, R. S. and Tecuci, G. (eds.). Morgan Kaufmann, Los Altos, CA. (Also available as Technical Report GIT-CC-92/19, College of Computing, Georgia Institute of Technology, Atlanta, GA).
- Ram, A., Cox, M. T. and Narayanan, S. (1992). An Architecture for Integrated Introspective Learning. In M. Weintraub (ed.), *Proceedings of the ML-92 Workshop on Computational Architectures for Supporting Machine Learning & Knowledge Acquisition*, Aberdeen, Scotland, (July 4).
- Ram, A., and Hunter, L. (1992). The Use of Explicit Goals for Knowledge to Guide Inference and Learning. *Applied Intelligence*, Vol 2, No. 1, pp. 47-73.
- Ram, A., and Leake, D. (1991). Evaluation of Explanatory Hypotheses. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. Chicago, IL, (August 7-10), pp. 867-871.
- Recker, M., and Pirolli, P. (to appear). Modelling Individual Differences in Student's Learning Strategies, *Journal of the Learning Sciences*.
- Redmond, M. A. (1992). "Learning by Observing and Understanding Expert Problem Solving." PhD Thesis, Technical Report GIT-CC-92/43, Georgia Institute of Technology, Atlanta, GA (September).
- Rosenbloom, P., Laird, J. and Newell, A. (1988). Meta-Levels in Soar. In P. Maes and D. Nardi (eds.), *Meta-Level Architectures and Reflection*. North Holland, Amsterdam, pp. 227-240.
- Schank, R. C. (1982). *Dynamic Memory: A theory of reminding and learning in computers and people*. Cambridge University Press, Cambridge, MA.
- Schank, R. C. (1986). *Explanation Patterns: Understanding mechanically and creatively*. Lawrence Erlbaum Associates, Hillsdale, NJ.
- Schank, R. C., and Leake, D. (1990). Creativity and Learning in a Case-Based Explainer. In J. G. Carbonell (ed.), *Machine Learning: Paradigms and methods*. MIT Press. Cambridge, MA.
- Schank, R. C., and Osgood, R. (1990). A Content Theory of Memory Indexing. Technical Report 2. Institute for the Learning Sciences, Northwestern University, Evanston, IL.
- Schank, R. C., and Owens, C. C. (1987). Understanding by Explaining Expectation Failures. In R. G. Reilly (ed.), *Communication Failure in Dialogue and Discourse*. Elsevier Science Publishers B. V., New York.
- Schneider, W. (1985). Developmental Trends in the Metamemory-Memory Behavior Relation-

- ship: An integrative review. In D. L. Forrest-Pressley, G. E. MacKinnon, and T. G. Waller (eds.), *Metacognition, Cognition and Human Performance*. Vol. 1 (Theoretical Perspectives), Academic Press, Inc., New York, pp. 57-109.
- Sleeman, D., Langley, P., and Mitchell, T. (1984). Learning from Solution Paths: An approach to the credit assignment problem. CIP Working Paper 443. Carnegie Mellon University. Pittsburgh, PA.
- Spear, N. E. (1978). *The Processing of Memories: Forgetting and retention*. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ.
- Stefik, M. (1981). Planning and Metaplanning (MOLGEN: Part 2), *Artificial Intelligence*. Vol. 16, pp. 141-169.
- Stroulia, E. (1992). "Towards a Functional Model of Reflective Learning." PhD Proposal, Technical Report GIT-CC-92/56, Georgia Institute of Technology, Atlanta, GA (November).
- Stroulia, E., and Goel, A. (1992). A Model-Based Approach to Incremental Self-Adaptation. In M. Weintraub (ed.), *Proceedings of the ML-92 Workshop on Computational Architectures for Supporting Machine Learning & Knowledge Acquisition*, Aberdeen, Scotland, (July, 4).
- Stroulia, E., Shankar, M., Goel, A., and Penberthy, L. (1992) A Model-Based Approach to Blame Assignment in Design. In J. S.Gero (ed.), *Proceedings of AID'92: Second International Conference on AI in Design*, (June), pp. 519-537.
- Suchman, L. (1987). *Plans and Situated Action: The problem of human-machine communication*. Cambridge Press. Cambridge, MA.
- Sussman, G. J. (1975). *A Computer Model of Skill Acquisition*. New York: American Elsevier.
- Van Lehn, K. (1991). Rule Acquisition Events in the Discovery of Problem Solving Strategies. *Cognitive Science*. Vol. 15, No. 1, pp. 1-47.
- Van Lehn, K., Jones, R. M., and Chi, M. T. H. (1992). A Model of the Self-Explanation Effect. *Journal of the Learning Sciences*, Vol. 2, No. 1, pp. 1-60.
- Veloso, M., and Carbonell, J. G. (1991). Automating Case Generation, Storage and Retrieval in PRODIGY, In R. S. Michalski, and G. Tecuci (eds.), *Proceedings of the First International Workshop on Multistrategy Learning*. Harpers Ferry, WV, (November), pp.363-377.
- Weinert, F. E. (1987). Introduction and Overview: Metacognition and motivation as determinants of effective learning and understanding. In F. E. Weinert and R. H. Kluwe (eds.), *Metacognition, Motivation, and Understanding*. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ.
- Weintraub, M. A. (1991). "An Explanation-Based Approach to Assigning Credit" PhD Thesis. Ohio State University.
- Wellman, H. M. (1983). Metamemory Revisited. In M. T. H. Chi (ed.), *Contributions to Human Development*. Vol. 9 (Trends in memory development research). S. Karger, AG, Basel, Switzerland.
- Wellman, H. M. (1985). The Origins of Metacognition. In D. L. Forrest-Pressley, G. E. MacKinnon, and T. G. Waller (eds.), *Metacognition, Cognition and Human Performance*. Vol. 1 (Theoretical Perspectives), Academic Press, Inc., New York, pp. 1-31.

Wellman, H. M., and Johnson, C. N. (1979). Understanding of Mental Process: A developmental study of "remember" and "forget," *Child Development*, Vol. 50, pp. 79-88.

Wilson, T. D., and Schooler, J. W. (1991). Thinking Too Much: Introspection can reduce the quality of preferences and decisions. *Journal of Personality and Social Psychology*, Vol. 60, No. 2, pp. 181-192.

Wisniewski, E. J., and Medin, D. L. (1991). Harpoons and Long Sticks: The interaction of theory and similarity in rule induction. In D. H. Fisher, M. J. Pazzani and P. Langley (eds.), *Concept Formation: Knowledge and experience in unsupervised learning*. Morgan Kaufmann Publishers, San Mateo, CA, pp. 237-278.

## Index

### A

abduction 2, 4, 20  
abstraction 32, 35, 53, 54  
analogy 2, 5, 20, 24, 36  
Anderson, J. R. 1  
AQUA 15, 17, 20, 33, 36, 38, 57  
assumptions 11-14, 26, 33, 54, 62

### B

background knowledge (BK) 1, 4-6, 12, 13, 21,  
32-36, 45, 49-53, 56, 60, 61, 66  
Banerji, R. 29, 32  
Bhatta, S. 31, 74  
Bielaczyc, K. 64  
Birnbaum, L. 6, 13, 21, 24, 40, 62, 63  
blame assignment, see credit assignment 3-6,  
20, 29, 36, 39, 40, 41, 43, 54, 62, 69, 70,  
72  
Booker, L. B. 32  
Buchanan, B. G. 21

### C

Callantine, T. J. 12  
Carbonell, J. G. 1, 3, 24, 36  
case-based planning 8, 43, 74  
case-based reasoning 9, 10, 20, 62  
CASTLE 62  
Chandrasekaran, B. 12  
CHEF 74  
Chi, M. T. H. 64, 65, 67  
Clancey, W. J. 12  
cognitive monitoring, see metacognition 65  
Collins, G. 13, 21, 24, 40, 62, 63  
comprehension 10, 60, 63-65, 68, 71, 72  
comprehension monitoring, see metacompre-  
hension  
constructive induction 2  
control knowledge 12  
control strategy learning 32  
Cox, M. T. 5, 10, 21, 26, 31, 40, 46, 52, 66, 72  
credit assignment, see blame assignment 4

### D

Davis, R. 21  
deception 29, 32  
Decide-Compute-Node (D-C-Node) 24, 25  
deciding what to learn 40, 41, 62, 66  
decision model 16, 22  
deduction 2  
DeJong, G. 31, 35  
derivational analogy 3, 24, 36  
Domeshek, E. A. 6, 12, 74  
Doyle, J. 48, 52

### E

EITHER 62  
elaborative question asking 31  
episodes 12  
evaluation, proposal 68-72  
explanation-based generalization (EBG) 10,  
31, 35, 41, 53, 56, 67  
explanation-based learning (EBL) 1  
explanation-based refinement 31  
explanation pattern, see XP

### F

failure, definition of 13  
failure taxonomy 26-30, 32, 44, 62  
Farrell, S. 74  
feeling of knowing 66  
Flavell, J. H. 65  
foreground knowledge (FK) 13, 17, 21, 52, 53,  
60, 61  
forget 5, 27, 45, 46, 51, 66  
Francis, A. 74  
Freed, M. 21, 40, 62, 63

### G

Garner, R. 65  
Gavelek, J. R. 63  
generalization 32, 54

generic task 12  
Goel, A. K. 12, 40, 62, 63  
Goldberg, D. E. 32

## H

Hammond, K. J. 13, 24, 31, 59, 74  
Hayes-Roth, B. 13, 24  
Hayes-Roth, F. 13, 24  
Hinrichs, T. R. 8, 9  
Holland, J. H. 32  
Hunter, L. E. 4, 31, 40, 43-45

## I

impasse 9, 13  
index learning 5, 10, 31, 35, 41, 50, 51, 53, 56, 67  
indexing problem 12, 26-28, 66  
induction 2, 32, 53  
Inferential Learning Theory (ILT) 1  
introspection 3, 5-7, 10, 15, 21, 33, 38, 58, 63, 68-72  
introspective explanation, see Meta-Explanation Pattern 23

## J

Johnson, C. N. 66  
Jones, R. M. 13, 64  
JULIA 8-10

## K

Kass, A. 36  
Kedar-Cabelli, S. 31, 35  
Keller, R. M. 31, 35, 40  
Knoblock, C. A. 1  
knowledge goal 20, 23, 24, 44, 45, 51, 56, 73  
Kolodner, J. L. 12, 13  
Krulwich, B. 21, 40, 62, 63  
Kuokka, D. R. 73

## L

Lachman, J. L. 66  
Lachman, R. 66  
Laird, J. E. 1, 72  
Langley, P. 32  
Leake, D. 20, 21, 36, 40  
learning goal 1, 3, 4, 6, 10, 11, 31, 36, 39, 41-44, 53, 62, 63, 69, 70-72  
LEX 29

## M

Maes, P. 21  
Mahesh, K. 74  
Medin, D. L. 1  
Meta-AQUA 6, 10, 32-38, 41, 43, 44, 48, 53, 56, 57, 62, 67, 69-72  
metacognition 63-67, 69, 71, 72  
metacomprehension 63, 65, 71  
Meta-Explanation Pattern (Meta-XP) 10, 21, 24, 26, 30, 33, 36, 38, 43, 56, 63, 66, 67  
metaknowledge 1, 13, 21, 65, 66  
meta-memory 65  
META-ROUTER 62  
Meta-XP theory 10, 14, 22, 63-68, 72  
Michalski, R. S. 1, 32, 40, 54  
MINERVA 62  
Minsky, M. L. 4, 40  
Minton, S. 1  
mistake 13  
Mitchell, T. 29, 31, 32, 35  
MOLGEN 14  
Mooney, R. 27, 31, 35, 62  
motivational explanation 22  
multistrategy learning 1-3, 10, 12, 43, 58, 62, 68, 72  
multistrategy reasoning 11, 12, 62

## N

Newell, A. 1, 5, 72

## O

opportunistic reasoning 13, 23, 24, 28, 32, 38,

42  
Osgood, R. 12  
Ourston, D. 27, 62  
Owens, C. 13, 14, 36, 59, 62, 63

## P

Park, Y. T. 62  
Pazzani, M. 70, 73  
Penberthy, L. 40  
Peterson, J. 74  
physical explanation 17, 23  
Pirulli, P. 64, 71  
PRODIGY 36  
Punch III, W. F. 12

## Q

question-driven understanding 15, 17-19, 33,  
34, 38, 49, 51, 61

## R

Ram, A. 5, 10, 13, 15, 16, 20-22, 24, 26, 31, 32,  
37, 40, 44, 46, 52, 66, 72  
Raphael, T. E. 63  
Recker, M. 71, 74  
recovery 59, 60  
Redmond, M. A. 43, 45  
repair 1, 6, 44, 59, 60, 62, 72  
Rosenbloom, P. S. 1, 72  
ROUTER 12

## S

Schank, R. C. 12-14, 21, 36  
Schneider, W. 65, 67  
Schooler, J. W. 70  
search-strategy learning 9  
self-explanation 64, 71  
Shankar, M. 40  
similarity-based learning 2, 10  
Simon, H. A. 5  
situated cognition 12  
Sleeman, D. 32

Soar 1, 71, 72  
Spear, N. E. 66  
Stefik, M. 14  
strategy selection 1, 3, 7, 12, 14, 20, 27, 40-42,  
58, 62, 70, 72  
Stroulia, E. 40, 63, 73, 74  
Suchman, L. 12  
SURF 71, 72  
surprise 13  
Sussman, G. J. 13  
SWALE 36, 38

## T

Tale-Spin 70, 71, 73  
Teccuci, G. 1  
Thronesbery, C. 66  
TIPS 12

## U

unexpected success 13  
Utgoff, P. E. 29, 32

## V

Van Lehn, K. 13, 64  
Veloso, M. 24, 36  
volitional explanation 17, 23

## W

Weinert, F. E. 67  
Weintraub, M. A. 40  
Wellman, H. M. 64-67  
Wilkins, D. C. 62  
Wilson, T. D. 70  
Wisniewski, E. J. 1

## X

XP (explanation pattern) 21, 35, 37, 56  
XP application 20, 36, 38, 39, 54