January 1987 UILU-ENG-87-2203

COORDINATED SCIENCE LABORATORY
College of Engineering

EXTENDING EXPLANATION-BASED LEARNING: FAILURE-DRIVEN SCHEMA REFINEMENT

Steve A. Chien

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UILU-ENG- 87-2203	5. MONITORING ORGANIZATION REPORT NUMBER(S) N/A
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Lab University of Illinois 6b. OFFICE SYMBOL (If applicable) N/A	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research
6c. ADDRESS (City, State and ZIP Code) 1101 W. Springfield Avenue Urbana, Illinois 61801	7b. ADDRESS (City, State and ZIP Code) 800 N. Quincy Street Arlington, VA 22217
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N-00014-86-K-0309
8c. ADDRESS (City, State and ZIP Code) 800 N. Quincy Street Arlington, VA 22217	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: N/A PROJECT NO.: N/A TASK NO.: N/A WORK UNIT NO.: N/A
11. TITLE (Include Security Classification) Extending Explanation-Based-Learning: Fail ure-Driven Schema Refinement	
12. PERSONAL AUTHOR(S) Steve A. Chien	
13a. TYPE OF REPORT Technical 13b. TIME COVERED FROM ____ TO ____	14. DATE OF REPORT (Yr., Mo., Day) January 1987 15. PAGE COUNT 17
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) explanation-based learning, incremental learning, schema refinement, natural language understanding, planning and counter-planning

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Current explanation-based learning systems assume domain theories that are computationally tractable. This paper describes a system being developed that refines schemata for use in narrative understanding, a domain in which a complete analysis of agent interactions is computationally intractable. This system employs an incremental approach that learns an initial schema using the assumption that other agents will not counter-plan (i.e. take actions that will interfere with the original planner's actions). However, when the system observes the failure of an actor's schema due to counter-planning by another agent, it refines the original schema. This is accomplished by indexing the counter-plan under the connecting causal chain to the original schema. This new knowledge allows the system to explain both similar failures and actions taken to prevent similar failures. This paper describes the need for incremental explanation-based learning and outlines an application of this approach to learning schemata for natural language processing.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code) 22c. OFFICE SYMBOL None

Technical Report
UILU-ENG-87-2203

EXTENDING EXPLANATION-BASED LEARNING : FAILURE-DRIVEN SCHEMA REFINEMENT*

Steve A. Chien

Artificial Intelligence Research Group
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801

January 1987

Abstract

This paper appears in the *Proceedings of the Third IEEE Conference on Artificial Intelligence Applications*, Orlando, Florida, February 1987.

Current explanation-based learning systems assume domain theories that are computationally tractable. This paper describes a system being developed that refines schemata for use in narrative understanding, a domain in which a complete analysis of agent interactions is computationally intractable. This system employs an incremental approach that learns an initial schema using the assumption that other agents will not counter-plan (i.e. take actions that will interfere with the original planner's actions). However, when the system observes the failure of an actor's schema due to counter-planning by another agent, it refines the original schema. This is accomplished by indexing the counter-plan under the connecting causal chain to the original schema. This new knowledge allows the system to explain both similar failures and actions taken to prevent similar failures. This paper describes the need for incremental explanation-based learning and outlines an application of this approach to learning schemata for natural language processing.

* This research was supported by the Office of Naval Research under grant N-00014-86-K-0309.

Extending Explanation-Based Learning:

Failure-Driven Schema Refinement

1. Introduction

Recently, a new approach to machine learning, *explanation-based learning* [DeJong81, DeJong86, Mitchell86], has attracted a great deal of attention. This approach differs greatly from earlier similarity-based techniques [Michalski83, Stepp86, Winston70] in that it utilizes a dependency structure (explanation) constructed using a domain theory in order to learn new concepts. Current EBL systems have shown considerable success in a wide range of application areas ranging from mathematical equation solving [Silver85], physics [Shavlik85], robotics [Segre85], integration problems [Mitchell83], circuit design [Mitchell85], and narrative processing [Mooney85].

However, existing EBL systems make certain simplifying assumptions about the domain theories that they use. In [Mitchell86], three classes of domain theory problems are described. First, the *incomplete theory problem* exists when the domain theory used by the learning system may not possess all of the information needed to properly explain observed events. One approach to dealing with the incomplete theory problem is described in [Rajamoney86]. With the *intractable theory problem*, a domain theory exists, but use of the theory to construct an exhaustive proof is not computationally tractable. Last, the *inconsistent theory problem* exists when the domain theory can derive conflicting facts. This paper describes a system, ARIES (for Automated Refinement and Indexing of Explanatory Schemata), which refines schemata for use in understanding narratives, a domain in which a complete analysis of agent interactions is computationally intractable.

2. Overview

ARIES is an extension to the GENESIS system [Mooney85]. The GENESIS system uses knowledge structures called schemata to process narratives. These schemata, similar to scripts, frames, or MOPs, represent prototypical knowledge about plans. GENESIS uses schemata to fill in input gaps which enables understanding of complex stories. This is done in a manner similar to [Cullingford78, DeJong82]. The GENESIS system improves its ability to process narratives by learning new schemata.

GENESIS learns plans for achieving *thematic goals*. Thematic goals [Schank77] are important goals which all agents are presumed to have (such as acquiring money, preserving one's freedom, etc.). When GENESIS observes an actor achieving a thematic goal in a manner not explained by one of its existing schemata and is able to explain how the agent's actions led to the achievement of the thematic goal, the system learns a schema describing the general method of achieving the goal. This schema contains both causal information (allowing GENESIS to understand narratives where the schema applies) and information on when to activate this schema. A schema is termed *active* if it is determined to be occurring in the narrative. For the purposes of this paper, we are concerned mainly with the causal description learned by GENESIS.

However, the causal description learned by GENESIS only connects facts supporting the achievement of the thematic goal. In certain cases, actions involved in the plan will motivate other agents to perform actions to prevent the successful execution of the original agent's plan (hereafter referred to as counter-planning [Carbonell79, Wilensky83]). Consider a plan to achieve money by robbing a bank. Clearly, a major portion of understanding how to rob a bank involves preventing the police, bank guards, and other possible counter-agents from interfering with your actions. Yet it is not computationally feasible to blindly determine all potential counter-agents and their possible counter-plans. Imagine a system enumerating all of the agents that it knows and determining whether they would be motivated to counter-plan. Such a system would require vast amounts of time and computing resources to understand the simplest of plans.

But people seem to have the ability to use their domain knowledge to predict counter-agents and certain counter-plans. This raises two important issues. The first is how people acquire the knowledge necessary to predict counter-agents and counter-plans. One way in which this counter-planning knowledge might be acquired is the approach used by ARIES - understanding and generalizing failures caused by counter-plans. The second issue is the amount of effort to devote to anticipation of novel counter-plans. While it seems that people are able to anticipate some unforseen counter-plans, the space of potential counter-plans and counter-agents is too large to search.

This research addresses the problem of learning plans involving interactions that are too complex to learn from a single example. ARIES uses thematic goal failures to indicate situations where schemata learned by GENESIS need to be augmented. This failure-driven approach to learning is similar to that discussed in [Schank82]. When ARIES sees a story in which actions taken by an agent in order to achieve a thematic goal also contribute to a thematic goal failure, it analyzes the failure in order to determine whether a relevant counter-plan should be learned. This knowledge can subsequently be used by ARIES to understand similar failures and to understand measures taken to prevent similar failures.

3. Failure-Driven Refinement

GENESIS accepts stories in English and uses an adaptation of McDypar [Dyer83] to parse them into a conceptual representation similar to predicate logic. When processing the conceptual representation, GENESIS attempts to connect the inputs causally into a *causal model*. ARIES monitors this causal model to determine which schemata need refinement. I term this process refinement because the system incrementally learns information about existing plans. After observing a failure, the system uses existing plan information and the example currently being processed to determine if the current knowledge needs to be augmented.

The detection and refinement process used by ARIES consists of four steps:

(1) *Failure Detection:* the system determines that an agent has had a thematic goal failure due to his execution of a known schema and that the thematic goal failure resulted from counter-planning.

(2) *Failure Explanation:* ARIES retrieves the explanation for the goal failure from the this causal model.

(3) *Failure Generalization:* the system generalizes the causal description of the events that led to the failure.

(4) *Failure Indexing:* ARIES indexes the failure under the original attempted plan.

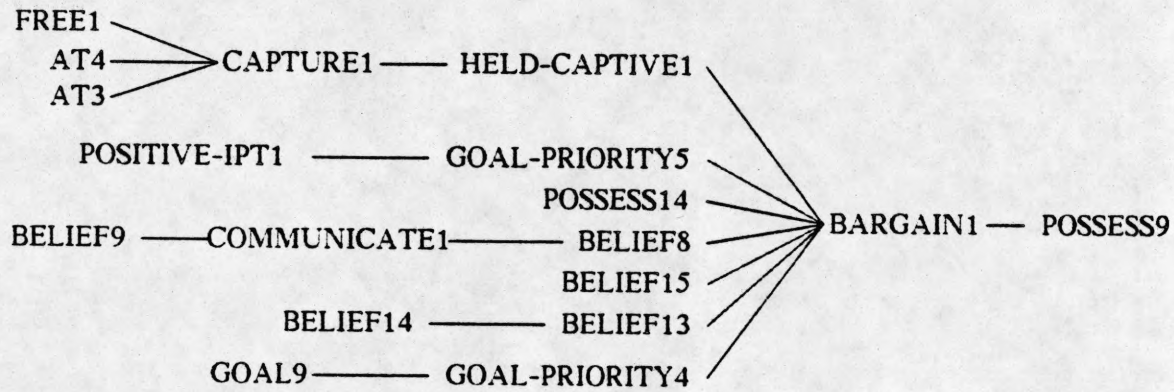
Although the system described refines only schemata representing plans (i.e. sequences of actions achieving goals), we feel that the approach outlined extends to object class descriptions. In this case failures would be incorrect predictions of functionality or object attributes due to proofs of class membership using simplifying assumptions. Because of the intended generality of the approach, I will use the term schema (which applies to both plans and class descriptions) and plan interchangeably.

Furthermore, although we are currently addressing failure descriptions corresponding to counter-plans, the approach outlined extends to learning general descriptions of failures. In this case the system would learn classes of interactions that were not investigated while understanding the original plan due to computational constraints.

I will now outline an example which will be used to illustrate the refinement process throughout the paper. Later, I will elaborate upon each of the steps involved in the schema refinement process. In the example, the system has already learned an initial description of kidnapping. The generalized explanation for the current kidnapping schema is shown in figure 1.

ARIES processes the conceptual representation for the following narrative:

> Nancy is the daughter of David, a wealthy businessman. Alan captured Nancy and locked her up in his lakeside cottage in eastern Illinois. Alan mailed a note to David stating that he would release Nancy if David gave him $50,000 at Coslow's. Alan got the money and Nancy was released.
>
> The next day the police captured Alan at his cottage. Nancy testified that Alan had abducted her. She had seen his face when he had captured her. Alan was convicted and sent to prison.



POSSESS9	Person1 has Money1.
BARGAIN1	Person1 makes a bargain with Person2 in which Person1 releases Person3 and Person2 gives Money1 to Person1.
POSSESS14	Person2 has Money1.
GOAL-PRIORITY5	Person2 wants Person3 free more than he wants to have Money1.
POSITIVE-IPT1	There is a positive interpersonal relationship between Person2 and Person3.
HELD-CAPTIVE1	Person1 is holding Person3 captive.
CAPTURE1	Person1 captures Person3.
FREE1	Person3 is free.
BELIEF8	Person2 believes Person1 is holding Person3 captive.
COMMUNICATE1	Person1 contacts Person2 and tells him that he is holding Person3 captive.
BELIEF9	Person1 believes he is holding Person3 captive.
BELIEF15	Person1 believes Person2 has Money1.
BELIEF13	Person1 believes Person2 wants Person3 to be free more that he wants to have Money1.
BELIEF14	Person1 believes there is a positive interpersonal relationship between Person2 and Person3.
GOAL-PRIORITY4	Person1 wants to have Money1 more than he wants to hold Person3 captive.
GOAL9	Person1 wants to have Money1
AT3	Person1 is at Location1
AT4	Person3 is at Location1

Figure 1: Original Kidnapping Schema

Using it's current description of kidnapping, GENESIS determines that Alan captured Nancy so that he could make a bargain with David in which David gave Alan $50,000 and Alan released Nancy. Also necessary for the bargain were that Alan wanted the money more than he wanted Nancy captive and that David wanted Nancy free more than he wanted the $50,000. Triggered by

Alan's loss of freedom (a thematic goal violation), ARIES begins the refinement process. Using it's domain knowledge, ARIES explains the failure as follows. Nancy saw Alan's face when he captured her; so she could identify him. Since Nancy was captured by Alan, she didn't like him. Consequently, she was willing to testify against him. Alan was held captive by the police, so he could be put on trial. As a result, Alan was convicted of a felony and sent to jail. The failure explanation is then generalized and indexed under the kidnapping plan as a counter-plan to avoid. The method used is elaborated in the following sections.

3.1. Detecting Plan Failures

The refinement process used by ARIES is triggered by thematic goal failures. While there are other cases in which refinement is desirable (e.g. recognition of a more efficient or more general action subsequence in a plan), we are currently only addressing certain plan failures as candidates for refinement. [DeJong83] gives criterion for deciding whether to learn a schema. Adapted to the context of learning failure schemata, these conditions are: 1) is a main goal of a character violated?; 2) is the violated goal a general one?; 3) does the input match an already existing schema?; 4) are the resources required by the counter-agents generally available? The first two criteria are satisfied because thematic goals are important goals that all agents are assumed to have. The third condition is satisfied by only triggering the refinement process when no existing failure schema explains the current input. The current work does not address the fourth condition.

In the kidnapping example, ARIES sees that Alan is no longer free, which violates his preserve-freedom goal. ARIES then attempts to explain the failure using an existing failure schema; if we already have a schema for this type of failure, clearly we don't want to waste the effort to relearn it. In this case, ARIES determines that no such schemata exists, and consequently proceeds with the refinement process. ARIES then notes that the failure is supported by components in the kidnap plan. First, the capture action in the kidnap plan supports Nancy's motivation for testifying. Second, the preconditions of the capture schema (that the location of the capturer be the same as the captured person) support Nancy seeing Alan's face. This is considered an interaction because the goal of capturing Nancy requires that Alan be at the same location as Nancy. Because the failure is supported by components in the kidnap plan, ARIES continues the refinement process.

More generally, when the system observes a thematic goal violation, it checks to see if it is explained by a currently active failure schema. If it is not explained by an active failure schema and it is caused in part by action(s) that are elements of an active plan then that active plan must be refined. Note that if there is an action in the original plan which has a precondition that supports the failure, there is a plan interaction (e.g. the second interaction in the example). This is because executing the plan causes those supports for the counter-plan.

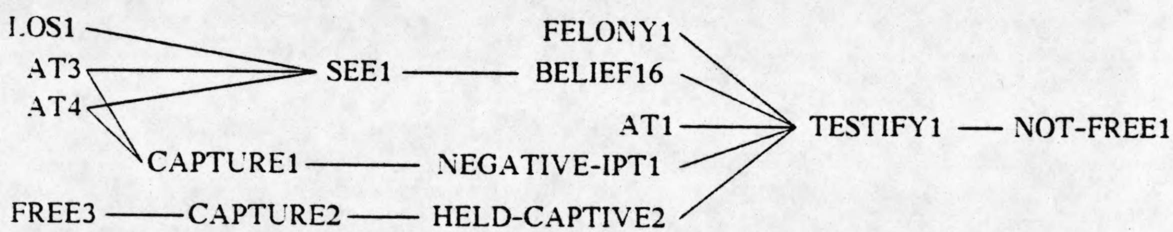
Even if ARIES cannot explain the original plan, much can be learned solely from the failure. Consider an example in which a someone commits a senseless murder (i.e. with no apparent motivation). The murderer is convicted because another person sees them commit the crime. The system learns a generalized failure schema where someone else observes the murder. Subsequently, the system processes an example in which someone attempts to murder their rich aunt in order to inherit their money and is caught because someone witnesses the murder. Clearly, the first failure schema will be useful in processing the second story. Thus, even if the original plan is not understood, ARIES can still learn from a failure example.

3.2. Explaining the Failure

As GENESIS receives inputs from the parser, the inferencer attempts to connect them with other facts believed by the system [Mooney85]. This is done in a backward-chaining fashion. If the input is an action, it attempts to find or infer the preconditions for the action. If the input is a state, there are two ways that the input can be connected. First, it could be connected by finding an applicable rule which asserts the state as a consequent. Alternatively, the system could infer a missing action that has the state as an effect. This process continues recursively until either the input is connected to previous information in the story, or a preset bound on system resources is exceeded. Because the amount of system resources devoted towards connecting inputs is limited, it is possible that the system may observe a thematic goal failure and not be able to explain why it occurred.

3.3. Generalizing the Failure

The failure is generalized in a manner equivalent to that used for successful schemata [Mooney85] using the EGGS system [Mooney86]. This system generalizes the actions causing the failure as far as possible without changing the validity of the explanation. The generalized explanation for the kidnapping failure is shown in Figure 2.



NOT-FREE1	Person1 is not free.
TESTIFY1	Person3 testifies that Person1 held Person3 captive.
FELONY1	Holding someone captive is a serious crime.
BELIEF16	Person3 believes that Person1 held Person3 captive.
AT1	Person3 is at the trial location.
NEGATIVE-IPT1	There is a negative interpersonal relationship between Person3 and Person1.
HELD-CAPTIVE2	The police are holding Person1 captive.
SEE1	Person3 sees Person1 capture Person3.
CAPTURE1	Person1 captures Person3.
CAPTURE2	The Police capture Person1.
LOS1	There is a clear line of sight from Person3 to Person1.
AT4	Person1 is at Location1.
AT3	Person3 is at Location1.
FREE3	Person1 is free.

Figure 2: Generalized Explanation for the Failure

3.4. Storing the Acquired Knowledge

Once we have a generalized explanation for the plan failure, it can now be indexed under the original schema as a possible failure. We do this by adding a counter-plan record to the kidnapping schema. This record contains a pointer to the capture-trial failure schema as well as pointers to the portions of both the original and the failure schemata that possibly have the interactions. This record gives us access to the following information about the failure: 1) the counter-agents responsible for the failure; 2) the counter-plan used; 3) the necessary states for the counter-plan to be motivated and executed (preconditions and motivations for the counter-plan). Later, we may add possible modifications to the original plan that block the counter-plan.

In the kidnapping example, a link is added from the kidnapping schema to the arrest-testify counter-plan. There are pointers to the AT3 and AT4 supports for the SEE1 and CAPTURE1 action and schema in the corresponding plan and counter-plan. The counter-agents are the Police and Person3 (the kidnap victim). The counter-plan used is the new arrest-testify schema. The preconditions and motivations are the lowest level supports for the arrest-testify schema: the victim and the kidnapper are at the same location, the victim has a clear line of sight to the kidnapper, that the kidnapper is free, and that the victim is at the trial location. Currently the system has no method to prevent this failure.

4. Using the Refined Schema

Once the counter-plan has been indexed under the original schema, the system can reason about the failure in two ways. First, ARIES can explain similar failures using the new counter-plan schema. Second, the information can be used to explain actions taken to prevent similar failures. Both of these methods involve recognizing that the counter-plan may be applicable in the current situation.

ARIES can use the associated failure schema to explain similar failures. If a similar failure is seen, the system can recognize that the failure schema is occurring. This allows the system to infer the causal structure stored with the failed schema and consequently understand why the agent failed.

ARIES can understand actions intended to prevent the failure from occurring. Previously, the only actions that would be understood are ones that appear in the causal explanation of the schema. However, now we may recognize actions taken to prevent counter-plans. This is done by determining if the particular action blocks a known counter-plan. For example, consider a kidnapping narrative where the kidnapper wears a mask while he is capturing the kidnap victim. This previously meaningless action can now be explained as the kidnapper preventing the capture-testify failure schema.

5. Comparisons to Previous Work

There has been considerable work on incremental approaches to machine learning. First, [Becker85, Michalski86, Reinke86] address how to learn incrementally using similarity-based techniques. However, these approaches did not deal with how to use domain knowledge to guide when and how to refine concepts. More applicable to our research are incremental approaches using domain knowledge by Pazzani, Hammond, and Doyle.

Pazzani's original work [Pazzani85] on the OCCAM system addressed similar issues in schema refinement in the context of memory organization. OCCAM used similarities to focus attention on situations where schemata need refinement. The OCCAM system learns a kidnapping schema as a specialization of the coercion TAU (an extremely high-level plan). OCCAM then observes a set of kidnapping examples where the victim is an infant. Directed by the similarity, it compares the infant examples to a failed plan in which the victim testifies against the kidnapper. Guided by the comparison to the planning failure, OCCAM produces the explanation that kidnapping infants prevents the victim from testifying against the kidnappers. The OCCAM system then indexes these types of plans as a specialization of the general kidnapping schema.

There are two major differences between OCCAM and ARIES. First, because OCCAM learns the kidnapping failure only as a specialization of kidnapping, it is not clear that knowledge of this failure could be applied to other relevant plans. Failures schemata in ARIES could be indexed under multiple relevant plans. The second difference is that OCCAM relies upon similarities to guide the explanation process while ARIES does not. This reliance upon similarities means that it cannot learn refinements from a single example.

Work by Hammond on the CHEF system [Hammond86a, Hammond86b] also addresses the problem of refining existing plans. The CHEF system plans for new problems by retrieving similar plans and modifying them to produce an initial plan. When a modified plan fails, CHEF first constructs an explanation for why the plan failed. CHEF then uses this explanation to classify the conditions that lead to the failure under a set of *planning TOPs*. The planning TOPs are general classifications of planning failures that suggest classes of actions to correct the failure. CHEF then uses heuristics to decide on a plan modification. CHEF also creates a rule using the generalized conditions that caused the failure in order to predict the failure. When subsequent plans are required which involve the same interaction that caused the failure, the modified plan will be retrieved due to the fact that it satisfies the newly predicted problem.

CHEF and ARIES differ in several important ways. First, CHEF presumes a powerful problem-solver that is able to solve anticipated problems. ARIES has no such problem-solving capabilities and consequently must learn a plan repair from observation. Furthermore, the problem solving component in CHEF uses a static classification of planning failures to suggest repairs (planning TOPS) which is assumed to be complete. No clear taxonomy of planning failures exists for diagnosing plans described in unrestricted narratives. Second, CHEF uses an episodic memory. This means that it can be determined whether a base-level plan causes a failure. In a schema based system, certain instantiations of a plan may cause failure, while other instantiations of the same plan will succeed. Finally, CHEF operates with complete knowledge of the plan. Because CHEF does the plan generation and execution itself, CHEF knows all of the steps in the plan as well as having access to all observable information during execution. When working from narratives describing plans, a system must be able to deal with missing plan steps, missing observable information, and irrelevant actions.

A third area of related work is Doyle's work in learning causal descriptions of devices [Doyle86]. In this approach, the system has several levels of detail in it's domain theory. The system uses this domain theory to explain the behavior of causal mechanisms. At the more detailed levels of the domain theory, the theory becomes more accurate. As predictions made by the current mechanism description are contradicted by observations made by the system, the system moves to a more detailed level of description.

Doyle's approach to model refinement is quite similar to that used by ARIES. However, Doyle's system uses schematic descriptions to explain behavior of mechanisms whereas ARIES builds the explanations from a more basic domain theory expressed in rules. This means that Doyle's approach depends on a predefined abstraction hierarchy to determine subsequent levels of refinement whereas our approach requires no such organization. Finally, Doyle's system learns causal descriptions of mechanisms where we are concerned mainly with understanding and refining plans.

6. Current Status

An initial version of the ARIES refinement system has been implemented [Chien86]. This system performed only the first three steps of the refinement algorithm described, and determined the preconditions of the failure schema. It then determined which of these preconditions were required by the original plan. The remaining preconditions were then added to the original plan as states to prevent. This initial system was not interfaced with the GENESIS system and did not use the EGGS generalization system [Mooney86].

The system currently being developed uses a more general learning scheme. The current approach allows for a more sophisticated explanation of how failures occurred and how they might be avoided. Additionally, the current version will use the EGGS generalization system and will be interfaced with the GENESIS system.

7. Future Work

There are several areas for future work. First, the refinement described here involves adding to the current plan description. While this in itself is important, it does not address the problem of how to actually change the explanation when it is incorrect. This is a complex problem because one must determine which portion of the current explanation is at fault (the credit assignment problem).

Another area for work is determining attempted execution of schemata. In the current system, the failed schemata must fail after the body of the original schema has occurred in order for GENESIS to determine that they have occurred. Yet imagine a situation in which a kidnapper allows the kidnap victim to escape, and is subsequently arrested. Currently, ARIES would not be able to recognize that the intended plan was kidnapping, and consequently could not learn that the kidnapper should attempt to prevent the victim from escaping.

A third area of research involves developing a better theory of "interestingness". Currently, we are treating only plans dealing directly with thematic goals as interesting. The only plans learned achieve thematic goals and the only failures addressed are those that involve thematic goals. Yet there are many worthwhile plans that do not directly lead to thematic goal fulfillments. And there are many important plan failures that do not involve thematic goal failures. Addressing these issues requires a much more powerful mechanism for determining whether a plan or failure is worth learning.

An additional area for work is indexing newly acquired failure schemata under schemata other than the original failure which caused it. If the system learns a capture-testify schema as a possible failure with kidnapping, it would be desirable to also realize that it would apply to any sort of criminal activity (i.e. murder, robbery, etc.). However, it is not immediately apparent how this might be done.

The most important area of research involves extending our approach to make general assumptions in initial plans and later refining these assumptions. This requires a problem-solver or understander that has the capability to reason about assumptions in general to determine which might be beneficial to make. Additionally, this system would need to be able to reason about these assumptions in light of later failures in order to determine which assumptions were not valid in the initial plan.

8. Conclusion

We have described an incremental approach to explanation-based learning despite the intractable domain theory problem. This approach involves learning an initial schema assuming that other agents will not counter-plan and using later failures to focus upon cases where this assumption does not hold. When the system observes a failure, it examines the causal explanation for the failure to determine if any currently active schemata need refinement. If an active schema supports the failure and ARIES does not possess a failure schema that already explains the failure, ARIES then generalizes the counter-plan into a failure schema. The original schema is then annotated with a record detailing the actions in the original schema that contributed to the failure and the preconditions of the failure enabled by the schema. With this knowledge, the system can explain actions taken by an agent to avoid a similar failure and explain similar failures.

Acknowledgements

This research work benefited from discussions with Ray Mooney. Suggestions and direction from my advisor, Gerald DeJong, and the other members of the CSL machine learning group also contributed to this research. Comments on an earlier draft of this paper by Gregg Collins and Brian Falkenhainer were also helpful.

References

[Becker85] J. Becker, "Topics in Incremental Learning of Discriminant Descriptions," Technical Report ISG 85-5, Intelligent Systems Group, University of Illinois at Urbana-Champaign, Urbana, IL, 1985.

[Carbonell79] J. G. Carbonell, "Subjective Understanding: Computer Models of Belief Systems," Technical Report 150, Ph.D. Thesis, Department of Computer Science, Yale University, New Haven, CT, 1979.

[Chien86] S. A. Chien, "A Failure-Driven Approach to Schema Refinement," Working Paper 81, Artificial Intelligence Research Group, Coordinated Science Laboratory, Urbana, IL, 1986.

[Cullingford78] R. E. Cullingford, "Script Application: Computer Understanding of Newspaper Stories," Technical Report 116, Department of Computer Science, Yale University, New Haven, CT, January 1978.

[DeJong81] G. F. DeJong, "Generalizations Based on Explanations," *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, B.C., Canada, August 1981, pp. 67-70. (Also appears as Working Paper 30, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[DeJong82] G. F. DeJong, "An Overview of the FRUMP System," in *Strategies for Natural Language Processing*, W. G. Lehnert and M. H. Ringle (ed.), Lawrence Erlbaum and Associates, Hillsdale, NJ , 1982.

[DeJong83] G. F. DeJong, "Acquiring Schemata through Understanding and Generalizing Plans," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, August 1983, pp. 462-464. (Also appears as Working Paper 56, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[DeJong86] G. F. DeJong and R. J. Mooney, "Explanation-Based Learning: An Alternative View," *Machine Learning 1*, 2 (April 1986), . (Also appears as Technical Report UILU-ENG-86-2208, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Doyle86] R. Doyle, "Constructing and Refining Causal Explanations from an Inconsistent Domain Theory," *Proceedings of the National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 538-544.

[Dyer83] M. J. Dyer, *In-Depth Understanding*, MIT Press, Cambridge, MA, 1983.

[Hammond86a] K. Hammond, "CHEF: A Model of Case-Based Planning," *Proceedings of the National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 267-271.

[Hammond86b] K. Hammond, "Learning to Anticipate and Avoid Planning Failures through the Explanation of Failures," *Proceedings of the National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 556-560.

[Michalski83] R. S. Michalski and R. E. Stepp, "Learning from Observation: Conceptual Clustering," in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, T. M. Mitchell (ed.), Tioga Publishing Company, Palo Alto, CA , 1983, pp. 331-363.

[Michalski86] R. S. Michalski, I. Mozetic, J. Hong and N. Lavrac, "The AQ15 Inductive Learning System: An Overview and Experiments," *Proceedings of the International Meeting on Advances in Learning*, Les Arcs, Switzerland, 1986.

[Mitchell83] T. M. Mitchell, P. E. Utgoff and R. Banerji, "Learning by Experimentation: Acquiring and Refining Problem-solving Heuristics," in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, T. M. Mitchell (ed.), Tioga Publishing Company, Palo Alto, CA , 1983, pp. 163-190.

[Mitchell85] T. M. Mitchell, S. Mahadevan and L. I. Steinberg, "LEAP: A Learning Apprentice for VLSI Design," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985, pp. 573-580.

[Mitchell86] T. M. Mitchell, R. Keller and S. Kedar-Cabelli, "Explanation-Based Generalization: A Unifying View," *Machine Learning 1*, 1 (January 1986), pp. 47-80.

[Mooney85] R. J. Mooney and G. F. DeJong, "Learning Schemata for Natural Language Processing," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985, pp. 681-687. (Also appears as Working Paper 67, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Mooney86] R. J. Mooney and S. W. Bennett, "A Domain Independent Explanation-Based Generalizer," *Proceedings of the National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, pp. 551-555 . (Also appears as Technical Report UILU-ENG-86-2216, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Pazzani85] M. J. Pazzani, "Explanation and Generalization Based Memory," *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, Irvine, CA, August 1985, pp. 323-328.

[Rajamoney86] S. A. Rajamoney, "Automated Design of Experiments for Refining Theories," M. S. Thesis, Department of Computer Science, University of Illinois, Urbana, IL, May 1986. (Also appears as Technical Report T-2213, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Reinke86] R. E. Reinke and R. S. Michalski, "Incremental Learning of Concept Descriptions," in *Machine Intelligence 11*, J. E. Hayes (ed.), Oxford University Press, Oxford, England, 1986.

[Schank77] R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*, Lawrence Erlbaum and Associates, Hillsdale, NJ, 1977.

[Schank82] R. C. Schank, *Dynamic Memory*, Cambridge University Press, Cambridge, England, 1982.

[Segre85] A. M. Segre and G. F. DeJong, "Explanation Based Manipulator Learning: Acquisition of Planning Ability Through Observation," *Proceedings of the IEEE International Conference on Robotics and Automation*, St. Louis, MO, March 1985, pp. 555-560. (Also appears as Working Paper 62, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Shavlik85] J. W. Shavlik, "Learning about Momentum Conservation," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985, pp. 667-669. (Also appears as Working Paper 66, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[Silver85] B. Silver, "Learning Control Information," *Proceedings of the 1985 International Machine Learning Workshop*, Skytop, PA, June 1985, pp. 189-191.

[Stepp86] R. E. Stepp, "Conceptual Clustering of Structured Objects," in *Machine Learning: A Guide To Current Research*, T. M. Mitchell, J. G. Carbonell and R. S. Michalski (ed.), Kluwer Academic Publishers, Hingham, MA, 1986, pp. 333-336.

[Wilensky83] R. W. Wilensky. *Planning and Understanding: A Computational Approach to Human Reasoning*. Addison-Wesley, Reading, MA, 1983.

[Winston70] P. H. Winston, "Learning Structural Descriptions from Examples," Technical Report 231, MIT AI Lab, Cambridge, MA, 1970.