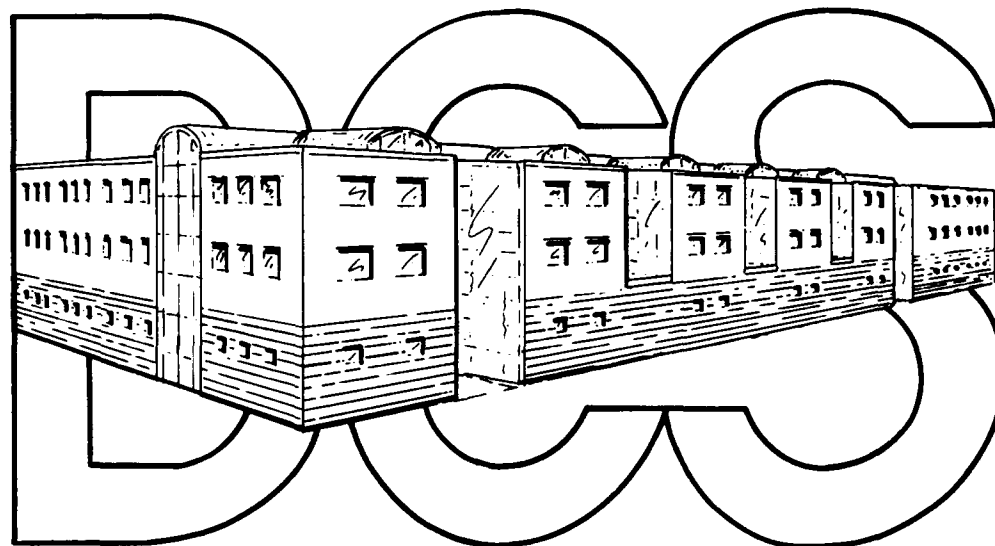


DTIC FILE COPY

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

AD-A224 441



THE NEW ADDITION

REPORT NO. UIUCDCS-R-90-1585

UIU-ENG-90-1724

**Knowledge Base Refinement as Improving
an Incorrect and Incomplete Domain Theory**

by

David C. Wilkins

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

April 1990

DTIC
SELECTE
JUL 30 1990
S & D

90 07 30 142

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution Unlimited	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UIUCDCS-R-90-1585 UILLU-ENG-90-1724		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION University of Illinois	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Artificial Intelligence (Code 1133) Cognitive Science (Code 1142CS)	
6c. ADDRESS (City, State, and ZIP Code) Dept. of Computer Science 1304 W. Springfield Ave. Urbana, IL 61801		7b. ADDRESS (City, State, and ZIP Code) Office of Naval Research 800 N. Quincy Street Arlington, VA 22217-5000	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-88K-0124	
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO 61153N	PROJECT NO. RR04206
		TASK NO OC	WORK UNIT ACCESSION NO 443g-008
11. TITLE (Include Security Classification) Knowledge Base Refinement As Improving An Incorrect and Incomplete Domain Theory			
12. PERSONAL AUTHOR(S) David C. Wilkins			
13a. TYPE OF REPORT Technical Report	13b. TIME COVERED FROM 87-12-1 TO 91-1-30	14. DATE OF REPORT (Year, Month, Day) 1990, April	15. PAGE COUNT 26
16. SUPPLEMENTARY NOTATION Available as Report UIUCDCS-R-90-1585, Dept. of Computer Science, University of Illinois, Urbana, IL 61801. Submitted to: Artificial Intelligence Journal .			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	artificial intelligence, machine learning, knowledge-based systems,	
05	09	knowledge base refinement, failure-driven learning, explanation-based learning,	
		apprenticeship learning, incomplete domain theory, incorrect domain theory.	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
The ODYSSEUS program automates knowledge base refinement by improving a domain theory. This paper describes the techniques used by ODYSSEUS to address three types of domain theory pathologies: incorrectness, inconsistency, and incompleteness. In ODYSSEUS, an incomplete domain theory is extended by the <i>Metarule Chain Completion Method</i>. This method exploits the use of an explicit meta-level representation of the strategy knowledge for a generic problem class (e.g., heuristic classification) that is separate from the domain theory (e.g., medicine) to be improved. Our work implements and compares the extension of an incomplete domain theory using case-based inductive learning and explanation-based apprenticeship learning; in the latter, learning occurs by completing failed explanations of observed human problem-solving actions. Extending an incomplete domain theory and correcting an incorrect domain theory both use the <i>Confirmation Decision Procedure Method</i>, which validates arbitrary instantiated tuples of the knowledge base by the use of an underlying domain theory. Lastly, the consistency of the knowledge base is improved by use of the <i>Sociopathic Reduction Algorithm</i>.			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Alan Meyrowitz, Dr. Susan Chipman		22b. TELEPHONE (Include Area Code) (202)696-4302, 696-4320	22c. OFFICE SYMBOL 1133 and 1142CS

Department of Computer Science
Report No. UIUCDCS-R-90-1585

College of Engineering
UIIU-ENG-90-1724

Knowledge Base Refinement As Improving An Incorrect and Incomplete Domain Theory

David C. Wilkins
Knowledge-Based Systems Group
Department of Computer Science
University of Illinois
405 North Mathews Ave
Urbana, IL 61801

April 1990

Submitted to:
Artificial Intelligence Journal
Earlier versions of this paper have appeared as
a Ph.D. Dissertation (1987), AAI-88 (1988), and MLC-89 (1989).

Title	
Author	☒
Editor	☐
Reviewer	☐
J. H. H. H.	
By	
Distribution	
Availability Codes	
Date	Available for
A-1	or



Knowledge Base Refinement as Improving an Incorrect and Incomplete Domain Theory

David C. Wilkins

Department of Computer Science
University of Illinois
405 North Mathews Ave
Urbana, IL 61801

Abstract

The ODYSSEUS program automates knowledge-base refinement by improving a domain theory. This paper describes the techniques used by ODYSSEUS to address three types of domain theory pathologies: incorrectness, inconsistency, and incompleteness.

In ODYSSEUS, an incomplete domain theory is extended by the *metarule chain completion method*. This method exploits the use of an explicit metalevel representation of the strategy knowledge for a generic problem class (e.g., heuristic classification) that is separate from the domain theory (e.g., medicine) to be improved. Our work implements and compares the extension of an incomplete domain theory using case-based inductive learning and explanation-based apprenticeship learning; in the latter, learning occurs by completing failed explanations of observed human problem-solving actions. Extending an incomplete domain theory and correcting an incorrect domain theory both use the *confirmation decision procedure method*, which validates arbitrary instantiated tuples of the knowledge base by the use of an underlying domain theory. Lastly, the consistency of the knowledge base is improved by use of the *sociopathic reduction algorithm*.

Contents

1	Introduction	3
2	MINERVA Classification and Design Shell	3
2.1	The Evolution from Heracles to Minerva	5
3	Odysseus's Method for Extending an Incomplete Domain Theory	7
3.1	Detection of Knowledge Base Deficiency	9
3.2	Suggesting a Knowledge Base Repair	10
3.3	Validation of Knowledge Base Repair	12
4	Odysseus's Method for Improving an Incorrect Domain Theory	12
5	Odysseus's Method for Improving an Inconsistent Domain Theory	14
6	Related Research	15
6.1	Odysseus and Explanation-Based Learning	15
6.2	Case-Based versus Apprenticeship Learning	15
7	Experimental Results	16
7.1	Improving an Incorrect and Inconsistent Domain Theory	17
7.2	Extending Incomplete Domain Theory via Case-Based Reasoning	18
7.3	Extending Incomplete Domain Theory via Apprenticeship Learning	18
8	Conclusions	19
9	Acknowledgements	21

1 Introduction

A central problem of expert systems is knowledge-base refinement (Buchanan and Shortliffe, 1984). Numerous research efforts have addressed the problem of improving an expert system that solves heuristic classification problems. The major research projects that have directly confronted this problem include the interactive semi-automatic approaches of TEIRESIAS (Davis, 1982), AQUINAS (Boose, 1984), and MORE (Kahn et al., 1985). They also include the automatic case-based inductive methods of INDUCE (Michalski et al., 1983), ID3 (Quinlan, 1983), SEEK2 (Ginsberg et al., 1985), and RL (Fu and Buchanan, 1985), which perform empirical induction over a library of test cases. This chapter describes a new approach to the refinement problem that involves a combination of failure-driven explanation-based learning and the use of underlying domain theories. Our approach is embodied in the ODYSSEUS learning program; ODYSSEUS contains specific (and separate) methods to address automatically three types of knowledge base pathologies: incorrectness, inconsistency, and incompleteness (Wilkins, 1987).

The remainder of this paper is organized as follows: Section 1.2 describes the MINERVA expert system shell that was specifically designed to facilitate failure-driven explanation-based learning. Our experience has shown that a sophisticated expert system architecture can provide an enormous amount of leverage to a learning program. Section 1.3 describes the apprenticeship learning methods used by ODYSSEUS to extend an incomplete domain theory; the key idea used to extend an incomplete domain theory is called the *metarule chain completion method*. Section 1.4 describes the methods used by ODYSSEUS to correct an incorrect domain theory; our approach to dealing with an incorrect domain theory is called the *confirmation decision procedure method*. Section 1.5 discusses the method used to remove certain types of inconsistencies from a correct but inconsistent domain theory; this method is called the *sociopathic reduction algorithm*. Section 1.6 presents results of a wide range of evaluation experiments that have been carried out, and Section 1.7 describes related research.

2 MINERVA Classification and Design Shell

The ODYSSEUS learning program can improve any knowledge base crafted for the MINERVA expert system shell; its overall organization is shown in Figure 1 (Park et al., 1989). MINERVA is a refinement of HERACLES, based on the experience gained in creating the ODYSSEUS apprenticeship learning program for HERACLES (Wilkins, 1987). HERACLES is itself

a refinement of EMYCIN, based on the experience gained in creating the GUIDON case-based tutoring program for EMYCIN (Clancey, 1986). These shells use a problem-solving method called *heuristic classification*, which is the process of selecting a solution out of a preenumerated solution set, using heuristic techniques (Clancey, 1985). The primary application knowledge base for MINERVA and HERACLES is the NEOMYCIN medical knowledge base for diagnosis of meningitis and similar neurological disorders (Clancey, 1984). This section describes the types of knowledge encoded in MINERVA and HERACLES, and how MINERVA differs from HERACLES.

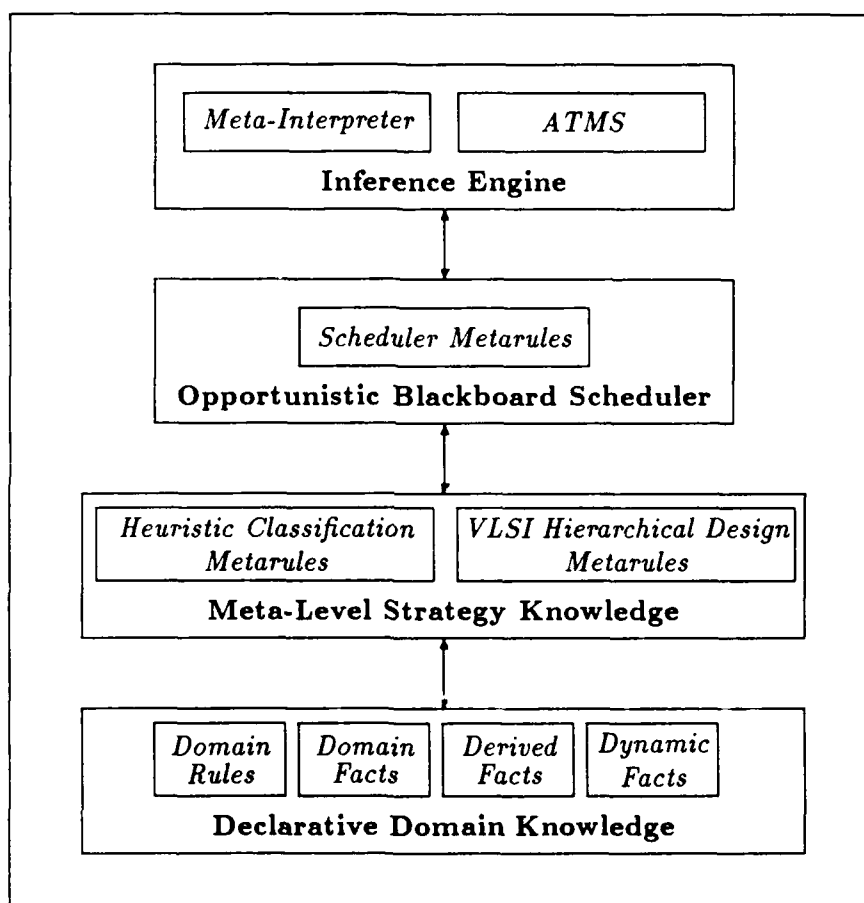


Figure 1: MINERVA System Architecture.

Domain knowledge consists of MYCIN-like rules and simple frame knowledge for an application domain such as medicine or geology. An example of rule knowledge in Horn clause format is `conclude(migraine-headache, yes, .5) :- finding(photophobia, yes)`, meaning 'to conclude the patient has a migraine headache with a certainty .5, determine if the patient has photophobia.' An example of frame knowledge is `subsumed-by(viral-meningitis, meningitis)`, meaning 'hypothesis viral meningitis is subsumed by the hypothesis meningitis.'

Problem-state knowledge is generated during execution of the expert system. Examples of problem-state knowledge are `rule-applied(rule163)`, which says that rule 163 has been applied during this consultation, and `differential(migraine-headache, tension-headache)`, which says that the expert system's active hypotheses are migraine headache and tension headache.

Strategy knowledge is contained in the shell, and it approximates a cognitive model of problem solving. For heuristic classification problems, this model is often referred to as hypothesis-directed reasoning (Elstein et al., 1978). The different problem-solving strategies that can be employed during problem solving are explicitly represented, which facilitates use of the model to follow the line of reasoning of a human problem solver. The strategy knowledge determines what domain knowledge is relevant at any given time, and what additional information is needed to solve the problem. The problem-state and domain knowledge, including rules, are represented as tuples; and strategy metarules are quantified over these tuples.

The strategy knowledge needs to access the domain and problem-state knowledge. To achieve this, the domain and problem-state knowledge is represented as tuples. Even rules are translated into tuples. For example, if rule 160 is `conclude(hemorrhage yes .5) :- finding(diplopia yes) ^ finding(aphasia yes)`, it would be translated into the following four tuples: `evidence.for(diplopia hemorrhage rule160 .5)`, `evidence.for(aphasia hemorrhage rule-160 .5)`, `antecedent(diplopia rule160)`, `antecedent(aphasia, rule160)`. Strategy metarules are quantified over the tuples. Figure 4 presents four strategy metarules in Horn clause form; the tuples in the body of the clause quantify over the domain and problem-state knowledge. The rightmost metarule in Figure 4 encodes the strategy to find out about a symptom by finding out about a symptom that subsumes it. The metarule applies when the goal is to find out symptom P1, and there is a symptom P2 that is subsumed by P1, and P2 takes Boolean values, and it is currently unknown, and P2 should be asked about instead of being derived from first principles. This is one of eight strategies in HERACLES that is also used in MINERVA for finding out the value of a symptom; this particular strategy of asking a more general question has the advantage of cognitive economy: a 'no' answer provides the answer to a potentially large number of questions, including the subsumed question.

2.1 The Evolution from Heracles to Minerva

MINERVA is a reworking of HERACLES, similar to the way that HERACLES is a reworking of EMYCIN. The ultimate objective in both these efforts has been a more declarative and modular representation of knowledge. This facilitates construction of a learning program to

examine and reason about the knowledge structures of the metalevel strategy in the expert system, to interpret better a user's strategy in terms of the metalevel strategy knowledge in the expert system, and to allow the same shell to encode strategy knowledge for the generic problem tasks of analysis (e.g., heuristic classification) and synthesis (e.g., VLSI circuit design).

There are four principal differences between MINERVA and HERACLES at the strategy level. In determining which task to perform next, HERACLES uses a fixed order goal tree; by contrast MINERVA employs an opportunistic blackboard scheduler. This facilitates interpreting a user's strategy in terms of the expert system's strategies, and better integrates top-down and bottom-up strategic reasoning. Second, in controlling metalevel reasoning, HERACLES uses dynamic control flags and variables, such as task end conditions. In MINERVA, a pure functional programming style and a deliberation-action loop have been used; this eliminates all flags and variables at the strategy level. So in MINERVA, the system state is completely determined by domain-level static and dynamic knowledge. Third, in HERACLES, strategy metarule premises sometimes change the state of the system, invoke subgoals, and use procedural attachment to LISP code; and HERACLES strategy metarule actions can invoke several goals. In contrast, MINERVA metarules do not follow any of these practices, which allows a pure deliberation-action cycle for strategic reasoning. The MINERVA style of metarules reduces side effects, thus making it easier for the learning program to reason about the strategy knowledge. Fourth, in MINERVA, more of the meta-level code in the expert system, such as the rule interpreter code, has been encoded in strategy metarules.

Other changes are as follows: The MINERVA system is completely implemented in PROLOG; by contrast, HERACLES uses a combination of PROLOG-like clauses with procedural attachment to LISP for each of the PROLOG clause predicates in metarules. The more uniform representation in MINERVA moves us toward our long-term goal of allowing a learning program to reason about all knowledge structures in the expert system shell. MINERVA incorporates an ATMS to maintain consistency of the knowledge base, uses a logic metainterpreter, and supports both certainty factors and Pearl's method to represent rule certainty and for propagation of information in a hierarchy of diagnostic hypotheses (Pearl, 1986b; Pearl, 1986a). As can be seen, all of the changes that mentioned have resulted in a more declarative and functional knowledge representation.

3 Odysseus's Method for Extending an Incomplete Domain Theory

We have developed two methods for extending an incomplete domain theory: an apprenticeship learning approach and a case-based reasoning approach. This section will only describe the former approach. Table 1 shows the major refinement steps and the method of achieving them for apprenticeship and case-based learning. The techniques will be elaborated below.

The solution approach of the ODYSSEUS apprenticeship program for extending an incomplete domain theory in a learning-by-watching scenario is illustrated in Figure 2. As Figure 2 shows, the learning process involves three distinct steps: detect domain theory deficiency, suggest domain theory repair, and validate domain theory repair. This section defines the concept of an explanation and then describes the three learning steps.

The main observable problem-solving activity in a diagnostic session is finding out values of features of the artifact to be diagnosed—we refer to this activity as asking *findout questions*. An *explanation* in ODYSSEUS is a proof that demonstrates how an expert's find-out question is a logical consequence of the current problem state, the domain and strategy knowledge, and one of the current high-level strategy goals. An explanation is created by backchaining the metalevel strategy metarules; Figure 4 provides examples of these metarules represented in Horn clause form. The backchaining starts with the findout metarule and continues until a metarule is reached whose head represents a high-level problem-solving

Learning Method	Case-Based Learning (similarity-based)	Apprenticeship Learning (explanation-based)
Scope	Heuristic rules.	Heuristic rules. 4 types of frame knowledge.
Detect KB deficiency	Select and run a case. Deficiency exists if case is misdiagnosed.	Observe expert solving a case. Deficiency exists if action of expert cannot be explained.
Suggest KB repair	Generalize or specialize rules. Induce new rules.	Find tuples that allow explanations to be completed under single fault assumption.
Validate KB repair	Use underlying domain theory to validate repairs.	Use underlying domain theory to validate repairs.

Table 1: Comparison of case-based and apprenticeship learning method for extending an incomplete domain theory.

goal. To backchain a metarule requires unification of the body of the Horn clause with domain and problem-state knowledge. Examples of high-level goals are: to test a hypothesis, to differentiate between several plausible hypotheses, to ask a clarifying question, and to ask a general question.

Apprenticeship learning is a form of learning by watching, in which learning occurs as a by-product of building explanations of human problem-solving actions. An apprenticeship is the most powerful method that human experts use to refine and debug their expertise in knowledge-intensive domains such as medicine. The major accomplishment of our method of apprenticeship learning is a demonstration of how an explicit representation of the strategy

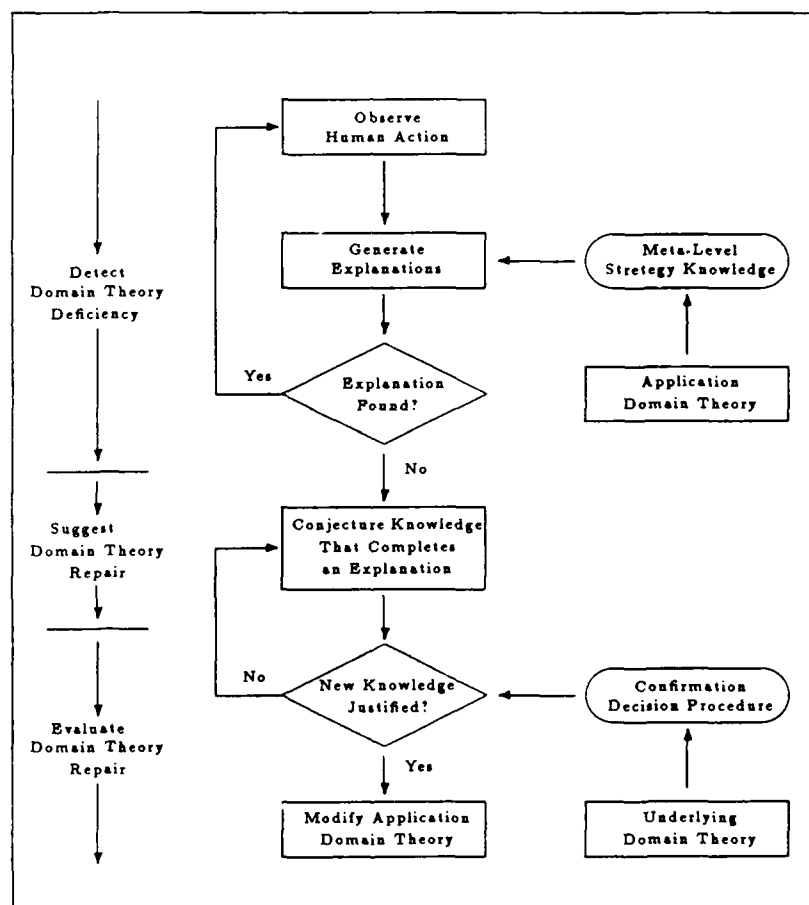


Figure 2: Overview of Odysseus' method for extending an incomplete domain theory in a learning-by-watching apprentice situation. This paper describes techniques that permit automation of each of the three stages of learning shown on the left edge of the figure. An explanation is a proof that shows how the expert's action achieves a problem-solving goal.

knowledge for a general problem class, such as diagnosis, can provide a basis for learning the knowledge that is specific to a particular domain, such as medicine.

3.1 Detection of Knowledge Base Deficiency

The first stage of learning involves the detection of a knowledge base deficiency. An expert's problem solving is observed and explanations are constructed for each of the observed problem-solving actions. An example will be used to illustrate our description of the three stages of learning, based on the NEOMYCIN knowledge base for diagnosing neurology problems. The input to ODYSSEUS is the problem-solving behavior of a physician, John Sotos, as shown in Figure 3. In our terminology, Dr. Sotos asks findout questions and concludes with a final diagnosis. For each of his actions, ODYSSEUS generates one or more explanations of his behavior.

When ODYSSEUS observes the expert asking a findout question, such as asking if the patient has visual problems, it finds all explanations for this action. When none can be

Patient's Complaint and Volunteered Information: 1. Alice Ecila, a 41 year old black female. 2. Chief complaint is a headache. Physician's Data Requests: 3. Headache duration? focus=tension headache. 7 days. 4. Headache episodic? focus=tension headache. No. 5. Headache severity? focus=tension headache. 4 on 0-4 scale. 6. Visual problems? focus=subarachnoid hemorrhage. Yes. 7. Double vision? focus=subarachnoid hemorrhage, tumor. Yes. 8. Temperature? focus=infectious process. 98.7 Fahrenheit. Physician's Final Diagnosis: 25. Migraine Headache.

Figure 3: An example of what the Odysseus apprentice learner sees. The data requests in this problem-solving protocol were made by John Sotos, M.D. The physician also provides information on the focus of the data requests. The answers to the data requests were obtained from an actual patient file from the Stanford University Hospital, extracted by Edward Herskovits, M.D.

found, an explanation failure occurs. This failure suggests that there is a difference between the knowledge of the expert and the expert system and it provides a learning opportunity. The knowledge difference may lie in any of the three types of knowledge that we have described: strategy knowledge, domain knowledge, or problem state knowledge. Currently, ODYSSEUS assumes that the cause of the explanation failure is that the domain knowledge is deficient. In the current example, no explanation can be found for findout question number 7 in figure 3 (asking about visual problems), and an explanation failure occurs.

3.2 Suggesting a Knowledge Base Repair

The second step of apprenticeship learning is to conjecture a knowledge base repair. A confirmation theory (which will be described in the discussion of the third stage of learning) can judge whether an arbitrary tuple of domain knowledge is erroneous, independent of the other knowledge in the knowledge base.

The search for the missing knowledge begins with the single fault assumption. It should be noted that the missing knowledge is described conceptually as a single fault, but because of the way the knowledge is encoded, we can learn more than one tuple when we

Group Hypotheses Strategy Metarule	Test Hypothesis Strategy Metarule	Applyrule Strategy Metarule	Findout Strategy Metarule
<code>goal(group-hyp(H1,H2))</code> <code>:- differential(H1),</code> <code>taxonomic(H1),</code> <code>parent(H2,H1),</code> <code>not pursued(H2),</code> <code>closest-common-</code> <code>ancestor(H2,H1),</code> <code>not(root(H2)),</code> <code>goal(test-hyp(H2)).</code>	<code>goal(test-hyp(H2)) :-</code> <code>concluded-by(H1,R1),</code> <code>not(pursued(R1)),</code> <code>inpremise(P1 R1),</code> <code>goal(applyrule(R1)).</code>	<code>goal(applyrule(R1)) :-</code> <code>not(rule-applied(R1)),</code> <code>inpremise(P1,R1),</code> <code>evid-for(P1,H2,R1 S1),</code> <code>soft-datum(P1),</code> <code>not(concluded(P1)),</code> <code>goal(findout(P1)),</code> <code>applyrule-forward(R1).</code>	<code>goal(findout(P1)) :-</code> <code>subsumes(P2,P1),</code> <code>not(concluded(P1)),</code> <code>boolean(P2),</code> <code>not(concluded(P2)),</code> <code>ask-user(P1).</code>

Figure 4: Learning by completing failed explanations. The illustrated strategy-level Horn clause metarules can chain together to form an explanation of how the the findout action of ask-user(P1) relates to the high-level goal of group-hypoth(H1,H2). In this particular case, all the tuples in the chain cannot be instantiated with domain knowledge. Odysseus' attempts to complete this and other failed explanation chains by adding domain knowledge to the knowledge base so that all the tuples unify.

learn rule knowledge. For ease of presentation, this feature is not shown in the following examples.

Conceptually, the missing knowledge could be eventually identified by adding a random domain knowledge tuple to the knowledge base and seeing whether an explanation of the expert's findout request can be constructed. How can a promising piece of such knowledge be effectively found? Our approach is to apply backward chaining to the findout question metarule, trying to construct a proof that explains why it was asked. When the proof fails, it is because a tuple of domain or problem-state knowledge needed for the proof is not in the knowledge base. If the proof fails because of problem-state knowledge, we look for a different proof of the findout question. If the proof fails because of a missing piece of domain knowledge, we temporarily add this tuple to the domain knowledge base. If the proof then goes through, the temporary piece of knowledge is our conjecture of how to refine the knowledge base.

Figure 4 illustrates one member of the set of failed explanations that ODYSSEUS examines in connection with the unexplained action, `ask-user(vizual problems)`, that is contained in the tail of the rightmost metarule. These strategy metarules create a chain between the high-level goal in the head of the leftmost metarule, `group-hypotheses(Hypothesis1, Hypothesis2)` and the low-level observable action in the tail of the rightmost metarule `ask-user(vizual problems)`. Note that this chain is but one path in a large explanation graph that connects the observable action of asking about vizual problems to all high-level goals. Each path in the graph is a potential explanation, and each node in a path is a strategy metarule. The failed explanation that ODYSSEUS is examining consists of the four metarules shown in Figure 4: Group Hypotheses, Test Hypothesis, Applyrule, and Findout. For a metarule to be used in a proof, its variables must be instantiated with domain or problem state tuples that are present in the knowledge base. In this example, the `evidence.for` tuple is responsible for the highlighted chain not forming a proof. It forms an acceptable proof if the tuple `evidence.for(photophobia acute.meningitis $rule $cf)` or `evidence.for(diplopia acute.meningitis $rule $cf)` is added to the knowledge base. During the step that generates repairs, neither the form of the left-hand side of the rule (e.g., number of conjuncts) or the strength is known. In the step to evaluate repairs, the exact form of the rule is produced in the process of evaluation of the worth of the tuple.

3.3 Validation of Knowledge Base Repair

The task of the third step of apprenticeship learning is to evaluate the proposed repair. To do this, we use the *confirmation decision procedure* (CDP) method. CDPs are constructed for each type of tuple in the domain theory, and can determine if the tuple is an acceptable tuple. Of the 19 different types of tuples in the Neomycin knowledge base, we have implemented CDPs for three of them: *evidence.for*, *clarifying.question*, and *ask.general.question* tuples. In addition to their use for validating knowledge base repairs, CDPs are also used to modify or delete incorrect parts of the initial domain theory; they are described in greater detail in Section 4.

Evidence.for tuples were generated in the visual problems example. In order to confirm the first candidate tuple, ODYSSEUS uses an empirical induction system that generates and evaluates rules that have photophobia in their premise and acute meningitis in their conclusion. A rule is found that passes the rule 'goodness' measures, and it is automatically added to the object-level knowledge base. All the tuples that are associated with the rule are also added to the knowledge base. This completes our example.

The CDP method also validates frame-like knowledge. An example of how this is accomplished will be described for *clarify question* tuples, such as *clarify.questions(headache-duration headache)*. This tuple means that if the physician discovers that the patient has a headache, she should always ask how long the headache has lasted. The confirmation theory must determine whether headache duration is a good clarifying question for the 'headache' symptom. To achieve this, ODYSSEUS first checks to see if the question to be clarified is related to many hypotheses (the ODYSSEUS explanation generator allows it to determine this), and then tests whether the clarifying question can potentially eliminate a high percentage of these hypotheses. If these two criteria are met, then the *clarify questions* tuple is accepted.

4 Odysseus's Method for Improving an Incorrect Domain Theory

The main focus of this chapter is on extending an incomplete domain theory via apprenticeship learning. However, it is clearly helpful if we are extending a domain theory that is correct and consistent. This section describes the methods that we have developed to improve the correctness of the domain theory. These methods are applied to the domain

theory prior to the use of apprenticeship learning.

The key to addressing the problem of incorrect knowledge is the use of the *confirmation decision procedure* (CDP) method, which connects tuples in the domain theory to underlying theories of the domain that are capable of judging their correctness. In this approach, a CDP is created for each type of domain theory tuple in the knowledge base. Given an arbitrary instantiated tuple, the CDP calculates whether the tuple is true or false. In some cases the CDP can suggest how the tuple can be modified so as to make it true.

Of the 19 different types of domain theory tuples in the NEOMYCIN domain theory, we have created CDP's for three types of tuples. These tuples comprise approximately 70% of all tuples in the domain theory. For example, a CDP has been implemented for *evidence.for* tuples. These tuples are derived from the heuristic domain rules provided by a user that relate evidence to hypotheses. Validating *evidence.for* tuples therefore consists of validating the heuristic associational rules in the knowledge base.

The CDP for *evidence.for* consists of an induction system, a set of rule biases, and a representative case library for the application domain. It accepts or rejects heuristic rules, whether they are rules in the initial knowledge base or rules conjectured during apprenticeship learning. In addition to accepting or rejecting rules, the CDP for *evidence.for* can modify a given rule to make it correct; it does this by adding conjuncts or modifying the rule strength. A rule can be modified to be "correct" by using probability and decision theory and representative sets of cases to determine its correct weight or strength (in contrast to trusting the weight provided by the user). If a rule lacks sufficient strength, the CDP will try to add conjuncts to the rule to increase its specificity.

When given an *evidence.for* tuple, its corresponding heuristic associational rule, which is indicated by the third argument of the *evidence.for* relation, is tested in five ways by the *evidence.for* CDP. A test for rule *simplicity* ensures that the number of antecedent conditions of the rules are less than the specified number. The test for *strength* accepts rules whose certainty factors (CF) are greater than a threshold value. The third bias is to test the *generality* of the rules. It succeeds only if the rules cover a certain percentage of the cases in a representative case library. The test for *colinearity* ensures that the proposed rules are not similar to any existing rules in performing classification of the induction set of cases. Finally the bias for *uniqueness* will check that the rules fire on a training case and there exist no rules in the current domain rule set that also succeed for that case. Good rules are those recommendations that pass the verification process. This rule may then be added into the system.

It is often difficult to create CDPs for some types of tuples in the domain theory. For example, consider the tuple type `askfirst(Parm)`. This tuple says that a particular feature of the system being diagnosed should be obtained from a user instead of derived from first-principles. It is difficult to imagine how to do this for an arbitrary feature, although eventually a way must be found if knowledge acquisition is to be completely automated.

Note that most knowledge bases are much more heterogeneous than LEAP, a learning apprentice for acquiring a domain theory that consists of VLSI circuit implementation rules. In this system, the domain theory only contains implementation rules (in our parlance, only contains one type of domain tuple). LEAP can verify the implementation rules using Kirchhoff's laws as its underlying domain theory. The challenge of using this idea for knowledge-base systems is that most domain theories contain many different types of domain knowledge, not just one type as in LEAP.

The CDPs were originally constructed to validate repairs during apprenticeship learning. However, they nicely allow the initial knowledge base to be validated prior to apprenticeship learning. As will be reported in Section 1.6, about half of the existing knowledge base is modified during the processing stage that focuses on ensuring that the domain theory contains correct knowledge.

5 Odysseus's Method for Improving an Inconsistent Domain Theory

A processing stage prior to apprenticeship learning also removes a form of inconsistent knowledge from the domain theory, which is responsible for deterioration of the performance of the system due to sociopathic interactions between elements of the domain theory. A domain theory is *sociopathic* if and only if (1) all the rules in the knowledge base individually meet some "goodness" criteria; and (2) a subset of the knowledge base gives better performance than the original knowledge base. The five biases described in Section 1.4 provide an example of goodness criteria for heuristic rules in the domain theory.

The significance of the phenomena of sociopaths is as follows. First, most extant expert systems have sociopathic knowledge bases. Second, traditional methods to correct missing and wrong rules, e.g., the general TEIRESIAS approach (Davis, 1982), cannot handle the problem. Third, sociopaths impose a limit on the quality of knowledge base performance. And last, it implies that some kind of global refinement for the acquired knowledge

is essential for machine learning systems.

The phenomena of sociopathicity is addressed at length in another paper, wherein we show that the best method for dealing with this form of inconsistency is to find a subset of the original domain theory that is not sociopathic (which must exist by our definition of sociopathicity). A summary of our results are as follows: The process of finding an optimal subset of a sociopathic knowledge base is modeled as a bipartite graph minimization problem and shown to be NP-hard. A heuristic method, the *sociopathic reduction algorithm*, has been developed to find a suboptimal solution for sociopathic domain theories. The heuristic method has been experimentally shown to give good results.

6 Related Research

6.1 Odysseus and Explanation-Based Learning

The ODYSSEUS apprenticeship learning method involves the construction of explanations, but it is different from explanation-based learning as formulated in EBG (Mitchell et al., 1986) and EBL (DeJong, 1986); it is also different from explanation-based learning in LEAP (Mitchell et al., 1989), even though LEAP also focuses on the problem of improving a knowledge-based expert system. In EBG, EBL, and LEAP, the domain theory is capable of explaining a training instance, and learning occurs by generalizing an explanation of the training instance. In contrast, in our apprenticeship research, a learning opportunity occurs when the domain theory, which is the domain knowledge base, is incapable of producing an explanation of a training instance. The domain theory is incomplete or erroneous, and all learning occurs by making an improvement to this domain theory.

6.2 Case-Based versus Apprenticeship Learning

In empirical induction from cases, a training instance consists of an unordered set of feature-value pairs for an entire diagnostic session and the correct diagnosis. In contrast, a training instance in apprenticeship learning is a single feature-value pair given within the context of a problem-solving session. This training instance is therefore more fine-grained, can exploit the information implicit in the order in which the diagnostician collects information, and allows obtaining many training instances from a single diagnostic session. Our apprenticeship learning program attempts to construct an explanation of each training instance; an

explanation failure occurs if none is found. The apprenticeship program then conjectures and tests modifications to the knowledge base that allow an explanation to be constructed. If an acceptable modification is found, the knowledge base is altered accordingly. This is a form of learning by completing failed explanations.

The case-based learning approach currently modifies or adds heuristic rules to the knowledge base. It runs all the cases in the library and locates those that are misdiagnosed. Given a misdiagnosed case, the local credit assignment problem is solved as follows: The premises of the rules that concluded the wrong final diagnosis are weakened by specialization, and the premises of the rules that concluded the correct diagnosis are strengthened. If this does not solve the problem, new rules will be induced from the patient case library that apply to the misdiagnosed case and that conclude the correct final diagnosis. The verification procedure used to test all knowledge base modifications is identical to that described for apprenticeship learning.

7 Experimental Results

Our knowledge-acquisition experiments centered on improving the ProHCD shell containing the NEOMYCIN knowledge base for diagnosing neurology problems. The initial NEOMYCIN knowledge base was constructed manually over a 7 year period; the first test of this system on a representative suite of test cases was performed in conjunction with the ODYSSEUS system. The NEOMYCIN vocabulary includes 60 diseases; our physician, Dr. John Sotos, determined that the existing data request vocabulary of 350 manifestations only allowed diagnosis of 10 of these diseases. Another physician, Dr. Edward Herskovits, constructed a case library of 115 cases for these 10 diseases from actual patient cases from the Stanford Medical Hospital, to be used for testing ODYSSEUS. The test set consisted of 112 of these cases.

Let us begin our performance analysis by considering the baseline system performance prior to any ODYSSEUS knowledge base refinement. The expected diagnostic performance that would be obtained by randomly guessing diagnoses is 10%, and the performance expected by always choosing the most common disease is 18%. Version 2.3 of HERACLES with the NEOMYCIN knowledge base initially diagnosed 31% of the cases correctly, which is 3.44 standard deviations better than always selecting the disease that is *a priori* the most likely. On a student t-test, this is significant at a $t = .001$ level of significance. Thus we can conclude that NEOMYCIN's initial diagnostic performance is significantly better than guessing. Version 3.1 of ProHCD, with the manually constructed NEOMYCIN knowledge

base, gave almost identical performance results; it initially diagnosed 32 of the 112 cases correctly (28.5% accuracy).

Table 1.2 shows the various diseases and their sample sizes in the evaluation set. The results of each test suite are described along three dimensions. TP (true-positive) refers to the number of cases that the expert system correctly diagnosed as present, FN (false-negative) to the number of times a disease was not diagnosed as present but was indeed present, and FP (false-positive) to the number of times a disease was incorrectly diagnosed as present.

Disease	Number Cases	KB1			KB2		
		TP	FN	FP	TP	FN	FP
Bacterial Meningitis	16	14	2	49	14	2	21
Brain Abscess	7	0	7	1	0	7	1
Cluster Headache	10	1	9	0	7	3	4
Fungal Meningitis	8	0	8	0	4	4	0
Migraine	10	4	6	6	1	9	0
Myco-TB Meningitis	4	0	4	2	4	0	0
Primary Brain Tumor	16	0	16	0	0	16	0
Subarach Hemorrhage	21	1	20	0	15	6	0
Tension Headache	9	7	2	5	7	2	6
Viral Meningitis	11	5	6	11	10	1	12
None	0	0	0	6	0	0	6
Totals	112	32	80	80	62	50	50

Table 2: Summary of MINERVA experiments. The KB1 column is the performance using the manually constructed domain theory. KB2 shows performance after use of methods that correct an incorrect domain theory.

7.1 Improving an Incorrect and Inconsistent Domain Theory

The first stage of improvement involves locating and modifying *incorrect* domain knowledge tuples. Our method modified 48% of the heuristic rules in the knowledge base. The improvement obtained using the refined knowledge base is shown in column KB2 of Table 1.2; ProHCD diagnosed 62 cases correctly (55.3% accuracy), showing an improvement of about 27%. The second stage of improvement involves correcting *inconsistent* domain knowledge. No experimental results are reported here, although our methods have been previously shown to lead to significant improvement (Wilkins and Ma, 1989).

<i>Disease</i>	<i>Number Cases</i>	KB3			KB4		
		TP	FN	FP	TP	FN	FP
Bacterial Meningitis	21	12	4	4	14	2	13
Brain Abscess	7	5	2	15	1	6	0
Cluster Headache	10	7	3	4	8	2	0
Fungal Meningitis	8	3	5	0	3	5	0
Migraine	10	4	6	0	6	4	0
Myco-TB Meningitis	4	4	0	0	4	0	1
Primary Brain Tumor	16	0	16	0	3	13	0
Subarach Hemorrhage	21	16	5	2	16	5	3
Tension Headache	9	7	2	6	8	1	3
Viral Meningitis	11	10	1	6	10	1	12
None	0	0	0	7	0	0	7
Totals	112	68	44	44	73	39	39

Table 3: Summary of MINERVA experiments. KB3 and KB4 show the performance after using case-based learning and apprenticeship learning, respectively, to extend the incomplete domain theory.

7.2 Extending Incomplete Domain Theory via Case-Based Reasoning

The third stage of improvement involves extending a correct but incomplete domain knowledge base. Two experiments were conducted. The first used case-based learning; all the cases were run, and two misdiagnosed cases in areas where the knowledge base was weak were selected. The case-based learning approach was applied to these two cases. This refinement, shown in column KB3 of Table 1.2, enabled the system to diagnose 68 cases correctly (60.7% accuracy), showing an aggregate improvement of 32%.

7.3 Extending Incomplete Domain Theory via Apprenticeship Learning

The second experiment used apprenticeship learning. For use as a training set, problem-solving protocols were collected by having Dr. Sotos solve two cases, consisting of approximately 30 questions each. ODYSSEUS discovered 10 pieces of knowledge by watching these two cases being solved; eight of these were domain rule knowledge. These eight pieces of information were added to the NEOMYCIN knowledge base of 152 rules, along with two pieces of frame knowledge that classified two symptoms as 'general questions'; these are questions that should be asked of every patient. This refinement, shown in column KB4

of Table 1.2, enabled the system to diagnose 73 cases correctly (65.2% accuracy), an aggregate improvement of about 37%. Compared to NEOMYCIN's original performance, the performance of NEOMYCIN after improvement by ODYSSEUS is 2.86 standard deviations better. On a student t-test, this is significant for $t = .006$. One would expect the improved NEOMYCIN to perform better than the original NEOMYCIN in better than 99 out of 100 sample sets.

It is important to note that the improvement occurred despite the physician's only diagnosing one of the two cases correctly. The physician correctly diagnosed a cluster headache case and misdiagnosed a bacterial meningitis case. As is evident from examining Tables 1.1 and 1.2, the improvement was over a wide range of cases, and the accuracy of diagnosing bacterial meningitis cases actually decreased. These counterintuitive results confirm our hypothesis that the power of our learning method derives from following the line of reasoning of a physician on individual findout questions and is not sensitive to the final diagnosis as is the case in learning by empirical induction from examples.

All of this new knowledge learned by apprenticeship learning was judged by Dr. Sotos as plausible medical knowledge, except for a domain rule linking aphasia to brain abscess. Importantly, the new knowledge was judged by our physician to be of much higher quality than when straight empirical induction was used to expand the knowledge base, without the use of explanation-based learning.

More experimental work remains. Our previous experiments with ODYSSEUS suggest that the apprenticeship learning approach is better than a case-based approach for producing a user-independent knowledge base to support multiple problem-solving goals such as learning, teaching, problem-solving, and explanation generation.

8 Conclusions

In this chapter, we presented the three distinct methods used by ODYSSEUS to improve a domain theory.

Our method of extending an incomplete domain theory is a form of failure-driven explanation-based learning, which we refer to as apprenticeship learning. Apprenticeship is the most effective means for human problem solvers to learn domain-specific problem-solving knowledge in knowledge-intensive domains. This observation provides motivation to give apprenticeship learning abilities to knowledge-based expert systems. The paradigmatic

example of an apprenticeship period is medical training, in which we have performed our investigations.

With respect to the incomplete theory problem, the research described illustrates how an explicit representation of the strategy knowledge for a general problem class, such as diagnosis, provides a basis for learning the domain-level knowledge that is specific to a particular domain, such as medicine, in an apprenticeship setting. Our approach uses a given body of strategy knowledge that is assumed to be complete and correct with the goal of learning domain-specific knowledge. This contrasts with learning programs such as LEX and LP where the domain-specific knowledge (e.g., integration formulas) is completely given at the start, and the goal of learning strategy knowledge (e.g., preconditions of operators) (Mitchell et al., 1983). Two sources of power of the ODYSSEUS approach are the method of completing failed explanations, called the *metarule chain completion method*, and the use of a underlying domain theories to evaluate domain-knowledge changes via the *confirmation decision procedure method*. Our approach complements the traditional method of empirical induction from examples for refining a knowledge base for an expert system for heuristic classification problems. With respect to learning certain types of heuristic rule knowledge, empirical induction from examples plays a significant role in our work. In these cases, an apprenticeship approach can be viewed as a new method of biasing selection of which knowledge is learned by empirical induction.

An apprenticeship learning approach, such as described in this chapter, is perhaps the best possible bias for automatic creation of a large 'user-independent' knowledge bases for expert systems. We desire to create knowledge bases that will support the multifaceted dimensions of expertise exhibited by some human experts, dimensions such as diagnosis, design, teaching, learning, explanation, and critiquing the behavior of another expert.

The long-term objectives of this research are the creation of learning methods that can harness an explicit representation of generic shell knowledge and that can lead to the creation of a user-independent knowledge base that rests on deep underlying domain models. Within this framework, this paper described specialized methods that address three major types of knowledge base pathologies: incorrect, inconsistent, and incomplete domain knowledge.

9 Acknowledgements

Many people have greatly contributed to the evolution of the ideas presented in this chapter. We would especially like to thank Bruce Buchanan, Bill Clancey, Tom Dietterich, Haym Hirsh, John Holland, John Laird, Pat Langley, Bob Lindsay, John McDermott, Ryszard Michalski, Roy Rada, Tom Mitchell, Paul Rosenbloom, Ted Shortliffe, Paul Scott, Devika Subramanian, Marianne Winslett, the members of the Grail learning group, and the Guidon tutoring group. This work would not have been possible without the help of physicians Eddy Herskovits, Kurt Kapsner, and John Sotos.

We would also like to express our deep gratitude to Lawrence Chachere, Ziad Najem, Young-Tack Park, and Kok-Wah Tan, and other members of the Knowledge-based Systems group at University of Illinois for their major role in the design and implementation of the MINERVA shell and for many fruitful discussions. This work was principally supported by NSF grant MCS-83-12148, and ONR grants N00014-79C-0302 and N00014-88K0124.

References

- Boose, J. H. (1984). Personal construct theory and the transfer of human expertise. In *Proceedings of the 1983 National Conference on Artificial Intelligence*, pages 27-33, Washington, D.C.
- Buchanan, B. G. and Shortliffe, E. H. (1984). *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Reading, Mass.: Addison-Wesley.
- Clancey, W. J. (1984). NEOMYCIN: Reconfiguring a rule-based system with application to teaching. In Clancey, W. J. and Shortliffe, E. H., editors, *Readings in Medical Artificial Intelligence*, chapter 15, pages 361-381. Reading, Mass.: Addison-Wesley.
- Clancey, W. J. (1985). Heuristic classification. *Artificial Intelligence*, 27:289-350.
- Clancey, W. J. (1986). From GUIDON to NEOMYCIN to HERACLES in twenty short lessons. *AI Magazine*, 7:40-60.
- Davis, R. (1982). Application of meta level knowledge in the construction, maintenance and use of large knowledge bases. In Davis, R. and Lenat, D. B., editors, *Knowledge-Based Systems in Artificial Intelligence*, pages 229-490. New York: McGraw-Hill.
- DeJong, G. (1986). An approach to learning from observation. In Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., editors, *Machine Learning, Volume II*, volume 2, chapter 19, pages 571-590. Los Altos: Morgan Kaufmann.
- Elstein, A. S., Shulman, L. S., and Sprafka, S. A. (1978). *Medical Problem Solving: An Analysis of Clinical Reasoning*. Cambridge: Harvard University Press.
- Fu, L. M. and Buchanan, B. G. (1985). Learning intermediate concepts in constructing a hierarchical knowledge base. In *Proceedings of the 1985 IJCAI*, pages 659-666, Los Angeles, CA.
- Ginsberg, A., Weiss, S., and Politakis, P. (1985). SEEK2: A generalized approach to automatic knowledge base refinement. In *Proceedings of the 1985 IJCAI*, pages 367-374, Los Angeles, CA.
- Kahn, G., Nowlan, S., and McDermott, J. (1985). MORE: An intelligent knowledge acquisition tool. In *Proceedings of the 1985 IJCAI*, pages 573-580, Los Angeles, CA.
- Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., editors (1983). *Machine Learning: An Artificial Intelligence Approach*. Palo Alto: Tioga Press.
- Mitchell, T., Utgoff, P. E., and Banerji, R. S. (1983). Learning by experimentation: Acquiring and refining problem-solving heuristics. In Michalski, T. M., Carbonell, J. G., and Mitchell, T. M., editors, *Machine Learning: An Artificial Intelligence Approach*, pages 163-190. Palo Alto: Tioga Press.
- Mitchell, T. M., Keller, R. M., and Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view. *Machine Learning*, 1(1):47-80.
- Mitchell, T. M., Mahadevan, S., and Steinberg, L. I. (1989). LEAP: A learning apprentice for VLSI design. In Kodratoff, Y. and Michalksi, R. M., editors, *Machine Learning III*, pages 000-000. Menlo Park: Morgan Kaufmann.

- Park, Y. T., Tan, K. W., and Wilkins, D. C. (1989). MINERVA: A knowledge based system shell with declarative representation and flexible control: For heuristic classification and VLSI design tasks. Technical Report UIUC-KBS-89-001, Department of Computer Science, University of Illinois, Urbana-Champaign, Illinois.
- Pearl, J. (1986a). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241-288.
- Pearl, J. (1986b). On evidential reasoning in a hierarchy of hypotheses. *Artificial Intelligence*, 28(1):9-15.
- Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end games. In Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., editors, *Machine Learning*, chapter 15, pages 463-482. Palo Alto: Tioga Press.
- Wilkins, D. C. (1987). *Apprenticeship Learning Techniques For Knowledge Based Systems*. PhD thesis, University of Michigan. Also, Report No. STAN-CS-88-1242, Dept. of Computer Science, Stanford University, 1988, 153pp.
- Wilkins, D. C. and Ma, Y. (1989). Sociopathic knowledge bases. Technical Report UIUCDCS-R-89-1538, Department of Computer Science, University of Illinois. Submitted to *Artificial Intelligence*.

ONR DISTRIBUTION LIST [ILLINOIS/WILKINS]

Ms. Lisa B. Achille
Code 5630
Naval Research Lab
Overlook Drive
Washington, DC 20375-5000

Dr. Thomas H. Andersen
Center for the Study of Reading
174 Children's Research Center
51 Gerty Drive
Champaign, IL 61820

Dr. Gautam Birwas
Department of Computer Science
Box 1688, Station B
Vanderbilt University
Nashville, TN 37235

Dr. Mark Burstein
BBN
10 Moulton Street
Cambridge, MA 02138

Dr. Edith Ackermann
Media Laboratory
E15-211 20 Ames Street
Cambridge, MA 02139

Dr. Stephen J. Andriole, Chairman
Department of Information Systems
and Systems Engineering
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr. John Black
Teachers College, Box 8
Columbia University
525 West 120th Street
New York, NY 10027

Dr. Robert Calfee
School of Education
Stanford University
Stanford, CA 94305

Dr. Philip Ackerman
Dept. of Psychology
University of Minnesota
75 East River road
N128 Elliott Hall
Minneapolis, MN 55455

Dr. John Annett
University of Warwick
Department of Psychology
Coventry CV4 7AL
ENGLAND

Dr. Daniel G. Bobrow
Intelligent Systems Laboratory
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

Dr. Robert L. Campbell
IBM T.J. Watson Research Center
P. O. Box 704
Yorktown Heights, NY 10598

Dr. Beth Adelson
Department of Computer Science
Tufts University
Medford, MA 02155

Dr. Edward Atkins
Code 81Z1210
Naval Sea Systems Command
Washington, DC 20362-5101

Dr. Deborah A. Boehm-Davis
Department of Psychology
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr. Joseph C. Campione
Center for the Study of Reading
University of Illinois
51 Gerty Drive
Champaign, IL 61820

Technical Document Center
AFHRL/LRS-TDC
Wright-Patterson AFB
OH 45433-8503

Dr. Michael E. Atwood
NYNEX
AI Laboratory
500 Westchester Avenue
White Plains, NY 10604

Dr. Sue Bogner
Army Research Institute
ATTN: PERI-SF
5001 Eisenhower Avenue
Alexandria, VA 22304-5000

Dr. Jaime G. Carbonell
Computer Science Department
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Robert Ahlers
Code N711
Human Factors Laboratory
Naval Training Systems Center
Orlando, FL 32813

Dr. Patricia Baggett
School of Education
610 E. University
University of Michigan
Ann Arbor, MI 48109-1259

Dr. Jeff Boner
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Thomas Carolan
Institute for Simulation and Training
University of Central Florida
12424 Research Parkway
Suite 300
Orlando, FL 32826

Dr. Robert M. Aiken
Computer Science Department
038-24
Temple University
Philadelphia, PA 19122

Dr. Bruce W. Ballard
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974

Dr. C. Alan Boneau
Department of Psychology
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr. Gail Carpenter
Center for Adaptive Systems
111 Commington St., Room 244
Boston University
Boston, MA 02215

Dr. Jan Aikins
AION Corporation
101 University
Palo Alto, CA 94301

Dr. Donald E. Bamber
Code 446
Naval Ocean Systems Center
San Diego, CA 92152-5000

Dr. J. C. Boudreaux
Center for Manufacturing
Engineering
National Bureau of Standards
Gaithersburg, MD 20899

Dr. John M. Carroll
IBM Watson Research Center
User Interface Institute
P.O. Box 704
Yorktown Heights, NY 10598

Dr. Saul Amarel
Dept. of Computer Science
Rutgers University
New Brunswick, NJ 08903

Dr. Harold Bamford
National Science Foundation
1800 G Street, N.W.
Washington, DC 20550

Dr. Lyle E. Bourne, Jr.
Department of Psychology
Box 245
University of Colorado
Boulder, CO 80309

CDR Robert Carter
Office of the Chief
of Naval Operations
OP-933D4
Washington, DC 20350-2000

Mr. Tejwansh S. Anand
Philips Laboratories
345 Scarborough Road
Briarcliff Manor
New York, NY 10520

Dr. Ranan Banerji
Dept. of Mathematics and CS
St. Joseph's University
5600 City Avenue
Philadelphia, PA 19131

Dr. Gary L. Bradshaw
Psychology Department
Campus Box 245
University of Colorado
Boulder, CO 80309

Dr. Fred Chang
Pacific Bell
2800 Camino Ramon
Room 2S-450
San Ramon, CA 94583

Dr. James Anderson
Brown University
Department of Psychology
Providence, RI 02912

Dipartimento di Psicologia
Via della Pergola 48
50121 Firenze
ITALY

kDr. Bruce Buchanan
Computer Science Department
University of Pittsburgh
322 Alumni Hall
Pittsburgh, PA 15260

Dr. Davida Charney
English Department
Penn State University
University Park, PA 16802

Dr. John R. Anderson
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Marie A. Bienkowski
333 Ravenswood Ave. PK327
SRI International
Menlo Park, CA 94025

LT COL Hugh Burns
AFHRL/IDI
Brooks AFB, TX 78235

Dr. Micheline Chi
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

ONR DISTRIBUTION LIST [ILLINOIS/WILKINS]

Dr. Susan Chipman
Personnel and Training Research
Office of Naval Research
Code 1142CS
Arlington, VA 22217-5000

Prof. Veronica Dahl
Department of Computer Science
Simon Fraser University
Burnaby, British Columbia
CANADA V5A 1S6

Dr. Ralph Dusek
V-P Human Factors
JIL Systems
1225 Jefferson Davis Hwy.
Suite 1209
Arlington, VA 22201

Dr. Jean-Claude Palmagne
Department of Psychology
New York University
8 Washington Place
New York, NY 10003

Dr. William J. Clancey
IRL
2550 Hanover Street
Palo Alto, CA 94304

Dr. John P. Dalphin
Chair, Computer Science Dept.
Towson State University
Baltimore, MD 21204

Prof. Michael G. Dyer
Computer Science Department
UCLA
3522 Boelter Hall
Los Angeles, CA 90024

Dr. Marshall J. Farr, Consultant
Cognitive & Instructional Sciences
2520 North Vernon Street
Arlington, VA 22207

Dr. Norman Cliff
Department of Psychology
Univ. of So. California
Los Angeles, CA 90089-1061

Dr. Charles E. Davis
Code 1142CS
800 N. Quincy Street
Arlington, VA 22217

Dr. John Ellis
Navy Personnel R&D Center
Code 51
San Diego, CA 92252

Dr. P.-A. Federico
Code 51
NPRDC
San Diego, CA 92152-8800

Dr. Paul Cohen
Computer Science Department
University of Massachusetts
Lederle Graduate Research Center
Amherst, MA 01003

Dr. Gerald F. DeJong
Dept. of Computer Science
University of Illinois
405 N. Mathews Ave.
Urbana, IL 61801

Dr. Susan Epstein
144 S. Mountain Avenue
Montclair, NJ 07042

Dr. Jerome A. Feldman
International Computer
Science Institute
1947 Center Street
Berkeley, CA 94704-1106

Dr. Alan Colling
BBN
33 Moulton Street
Cambridge, MA 02238

Dr. Thomas E. DeZern
Project Engineer, AI
General Dynamics
PO Box 748/Mail Zone 2646
Fort Worth, TX 76101

ERIC Facility-Acquisitions
4350 East-West Hwy., Suite 1100
Bethesda, MD 20814-4476

Dr. Paul Feltovich
Southern Illinois University
School of Medicine
Medical Education Department
P.O. Box 19230
Springfield, IL 62708

Dr. Stanley Collyer
Office of Naval Technology
Code 222
800 N. Quincy Street
Arlington, VA 22217-5000

Dr. Thomas G. Diesterich
Dept. of Computer Science
Oregon State University
Corvallis, OR 97331

Dr. K. Anders Ericsson
University of Colorado
Department of Psychology
Campus Box 245
Boulder, CO 80309-0245

Dr. Richard Fikes
Price Waterhouse Tech Center
68 Willow Road
Menlo Park, CA 94025

Dr. Greg Cooper
Stanford University
Knowledge Systems Lab
P. O. Box 8070
Stanford, CA 94305

Dr. Ronna Dillon
Department of Guidance and
Educational Psychology
Southern Illinois University
Carbondale, IL 62901

Dr. Lee Erman
Technowledge, Inc.
525 University Avenue
Palo Alto, CA 94301

CAPT J. Finelli
Commandant (G-PTE)
U.S. Coast Guard
2100 Second St., S.W.
Washington, DC 20592

Dr. Richard L. Coulson
Dept. of Physiology
School of Medicine
Southern Illinois University
Carbondale, IL 62901

Dr. J. Stuart Donn
Faculty of Education
University of British Columbia
2125 Main Mall
Vancouver, BC
CANADA V6T 1Z5

Dr. Tom Eskridge
Lockheed Austin Division
6800 Burleson Road
Dept. T4-41, Bldg. 30P
Austin, TX 78744

Dr. Douglas Fisher
Dept. of Computer Science
Box 67, Station B
Vanderbilt University
Nashville, TN 37235

Dr. Meredith P. Crawford
3582 Hamlet Place
Chevy Chase, MD 20815

Dr. Kejitan Dantas
George Mason University
Dept. of Computer Science
4400 University Drive
Fairfax, VA 22030

Dr. Lorraine D. Eyde
Office of Personnel Management
Office of Examination Development
1900 E St., NW
Washington, DC 20415

Dr. Donald Fitzgerald
University of New England
Department of Psychology
Armidale, New South Wales 2351
AUSTRALIA

Dr. Hans P. Crombag
Faculty of Law
University of Limburg
P.O. Box 616
Maastricht
The NETHERLANDS 6200 MD

Defense Technical
Information Center
Cameron Station, Bldg 5
Alexandria, VA 22314
Attn: TC
(12 Copies)

LCDR Micheline Y. Eyraud
Code 602
Naval Air Development Center
Warminster, PA 18974-5000

Mr. Nicholas S. Plann
Dept. of Computer Science
Oregon State University
Corvallis, Oregon 97331-3902

Dr. Kenneth B. Cross
Anacapa Sciences, Inc.
P.O. Drawer Q
Santa Barbara, CA 93102

Dr. Pierre Duguet
Organisation for Economic
Cooperation and Development
2, rue Andre-Pascal
75016 PARIS
FRANCE

Prof. Lawrence M. Pagan
Stanford University Medical Center
TC-135
Medical Computer Science
Stanford, CA 94305

Dr. Kenneth D. Forbus
Department of Computer Science
University of Illinois
405 N. Mathews Avenue
Urbana, IL 61801

Dr. Cary Cuijchen
Intelligent Instructional Systems
Texas Instruments AI Lab
P.O. Box 660246
Dallas, TX 75266

Dr. Brian Falkenhainer
Xerox PARC
3232 Coyote Hill Rd.
Palo Alto, CA 94304

Dr. Kenneth M. Ford
Division of Computer Science
The University of West Florida
11000 University Parkway
Pensacola, FL 32514

ONR DISTRIBUTION LIST [ILLINOIS/WILKINS]

Dr. Charles Forgy
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Barbara A. Fox
University of Colorado
Department of Linguistics
Boulder, CO 80309

Dr. Mark Fox
Carnegie Mellon University
Robotics Institute
Pittsburgh, PA 15213

Dr. Carl H. Frederiksen
Dept. of Educational Psychology
McGill University
3700 McTavish Street
Montreal, Quebec
CANADA H3A 1Y2

Dr. John R. Frederiksen
BBN Laboratories
10 Moulton Street
Cambridge, MA 02238

Dr. Norman Frederiksen
Educational Testing Service
(05-R)
Princeton, NJ 08541

Dr. Alfred R. Fregly
AFOSR/NL, Bldg. 410
Bolling AFB, DC 20332-8448

Dr. Peter Friedland
Chief, AI Research Branch
Mail Stop 244-17
NASA Ames Research Center
Moffett Field, CA 94035

Dr. Michael Friendly
Psychology Department
York University
Toronto ONT
CANADA M2J 1P3

Col. Dr. Ernst Frise
Heerespsychologischer Dienst
Maria Theresien-Kaserne
1130 Wien
AUSTRIA

Dr. Robert M. Gagne
1454 Mitchell Avenue
Tallahassee, FL 32303

Dr. Brian R. Gaines
Knowledge Science Institute
University of Calgary
Calgary, Alberta
CANADA T2N 1N4

Dr. Dedre Gentner
Department of Psychology
University of Illinois
603 E. Daniel
Champaign, IL 61820

Dr. Donald R. Gentner
Philips Laboratories
345 Scarborough Road
Briarcliff Manor, NY 10510

Dr. Helen Gigley
National Science Foundation
1800 G Street N.W.
Room 304
Washington, DC 20550

Dr. Philip Gillis
Army Research Institute
PERI-II
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Allen Ginsberg
AT&T Bell Laboratories
Holmdel, NJ 07733

Mr. Lee Gladwin
305 Davis Avenue
Leesburg, VA 22075

Dr. Robert Glaser
Learning Research
& Development Center
University of Pittsburgh
3920 O'Hara Street
Pittsburgh, PA 15260

Dr. Marvin D. Glock
101 Homestead Terrace
Ithaca, NY 14856

Dr. Dwight J. Goehring
ARI Field Unit
P.O. Box 5787
Presidio of Monterey
CA 93944-5011

Dr. Joseph Goguen
Computer Science Laboratory
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Mr. Richard Golden
Psychology Department
Stanford University
Stanford, CA 94305

Mr. Harold Goldstein
University of DC
Department Civil Engineering
Bldg. 42, Room 112
4200 Connecticut Avenue, N.W.
Washington, DC 20008

Dr. Sherrie Gott
AFHRL/MOMJ
Brooks AFB, TX 78235-5601

Dr. T. Govindaraj
Georgia Institute of
Technology
School of Industrial
and Systems Engineering
Atlanta, GA 30332-0206

Dr. Art Graesser
Dept. of Psychology
Memphis State University
Memphis, TN 38152

Dr. Wayne Gray
Artificial Intelligence Laboratory
NYNEX
500 Westchester Avenue
White Plains, NY 10604

H. William Greenup
Dep Asst C/S, Instructional
Management (E03A)
Education Center, MCCDC
Quantico, VA 22134-5050

Dr. Dik Gregory
Admiralty Research
Establishment/AXB
Queens Road
Teddington
Middlessex, ENGLAND TW110LN

Dr. Gleen Griffin
Naval Education and Training Program
Management Support Activity
Instructional Technology Impl. Div.
Code 0473
Pensacola, FL 32509-5000

Dr. Benjamin N. Grosz
IBM T.J. Watson Labs
P.O. Box 704
Yorktown Heights, NY 10598

Dr. Stephen Grossberg
Center for Adaptive Systems
Room 244
111 Cummington Street
Boston University
Boston, MA 02215

Michael Haben
DORNIER GMBH
P.O. Box 1420
D-7890 Friedrichshafen 1
WEST GERMANY

Dr. Henry M. Half
Half Resources, Inc.
4918 33rd Road, North
Arlington, VA 22207

Dr. H. Hamburger
Department of Computer Science
George Mason University
Fairfax, VA 22030

Dr. Cheryl Hamel
Naval Training Systems Center
Code 712
12250 Research Parkway
Orlando, FL 32826

Dr. Bruce W. Hamill
Research Center
The Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Road
Laurel, MD 20707

Dr. Chris Hammond
Dept. of Computer Science
University of Chicago
1100 E. 58th Street
Chicago, IL 60637

Dr. Patrick R. Harrison
Computer Science Department
U.S. Naval Academy
Annapolis, MD 21402-5002

Dr. Peter Hart
301 Arber Road
Menlo Park, CA 94025

Dr. Wayne Harvey
Center for Learning Technology
Education Development Center
55 Chapel Street
Newton, MA 02160

Dr. David Haussler
402 Nobel Drive
Santa Cruz, CA 95060

Dr. Barbara Hayes-Roth
Knowledge Systems Laboratory
Stanford University
701 Welch Road
Palo Alto, CA 94304

Dr. Frederick Hayes-Roth
Teknowledge
P.O. Box 10119
1850 Embarcadero Rd.
Palo Alto, CA 94303

Dr. James Hendler
Dept. of Computer Science
University of Maryland
College Park, MD 20742

Dr. James Hiebert
Department of Educational
Development
University of Delaware
Newark, DE 19716

Dr. Geoffrey Hinton
Computer Science Department
University of Toronto
Sandford Fleming Building
10 King's College Road
Toronto, Ontario
CANADA M5S 1A4

ONR DISTRIBUTION LIST [ILLINOIS/WILKINS]

Dr. Haym Hirsh
Dept. of Computer Science
Rutgers University
New Brunswick, NJ 08903

Mr. Roland Jones
Mitre Corp., K-203
Burlington Road
Bedford, MA 01730

Dr. Wendy Kellogg
IBM T. J. Watson Research Ctr.
P.O. Box 704
Yorktown Heights, NY 10598

Dr. Gary Kress
628 Spenser Avenue
Pacific Grove, CA 93958

Dr. James E. Hoffman
Department of Psychology
University of Delaware
Newark, DE 19711

Prof. Aravind K. Joshi
Department of Computer Science
University of Pennsylvania
R-208 Moore School
Philadelphia, PA 19104

Dr. Douglas Kelly
University of North Carolina
Department of Statistics
Chapel Hill, NC 27514

Prof. Casimir A. Kulikowski
Department of Computer Science
Hill Center
for the Mathematical Sciences
Busch Campus
Rutgers University
New Brunswick, NJ 08903

Dr. John H. Holland
Dept. of EE and CS
Room 2118
University of Michigan
Ann Arbor, MI 48109

Dr. Gary Kahn
1220 Macon Avenue
Pittsburgh, PA 15218

Dr. J.A.S. Kelso
Center for Complex Systems
Building MT 9
Florida Atlantic University
Boca Raton, FL 33431

Dr. David R. Lambert
Naval Ocean Systems Center
Code 772
271 Catalina Boulevard
San Diego, CA 92152-5000

Ms. Julia S. Hough
110 W. Harvey Street
Philadelphia, PA 19144

Dr. Ruth Kanfer
University of Minnesota
Department of Psychology
Elliott Hall
74 E. River Road
Minneapolis, MN 55455

Prof. Larry Kerschberg
Dept. of Information System
& Systems Engineering
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr. Pat Langley
NASA Ames Research Center
Mail Stop 244-17
Moffett Field, CA 94035

Dr. Jack Hunter
2122 Coolidge Street
Lansing, MI 48906

Dr. Michael Kaplan
Office of Basic Research
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Dennis Kibler
Dept. of Information & Computer Science
University of California
Irvine, CA 92717

Dr. Robert W. Lawler
Matthews 118
Purdue University
West Lafayette, IN 47907

Dr. Ed Hutchins
Intelligent Systems Group
Institute for
Cognitive Science (C-015)
UCSD
La Jolla, CA 92093

Mr. Shyam Kapur
Dept. of Computer Science
Cornell University
4120 Upson Hall
Ithaca, NY 14853

Dr. David Kieras
Technical Communication Program
TIDAL Bldg. 2360 Bonisteel Blvd.
University of Michigan
Ann Arbor, MI 48109

Dr. Yuh-Jeng Lee
Department of Computer Science
Code 52
Naval Postgraduate School
Monterey, CA 93942

Dr. Wayne Iba
Dept. of Information and CS
University of California, Irvine
Irvine, CA 92717

Dr. Demetrios Karis
GTE Labs, MS 61
40 Sylvan Road
Waltham, MA 02254

Dr. Thomas Killion
AFHRL/OT
Williams AFB, AZ 85240-8457

Ms. Debbie Leishman
Knowledge Science Institute
University of Calgary
Calgary, Alberta
CANADA T2N 1N4

Dr. Robin Jeffries
Hewlett-Packard Laboratories, 3L
P.O. Box 10490
Palo Alto, CA 94303-0971

Dr. A. Karmiloff-Smith
MRC-CDU
17 Gordon Street
London
ENGLAND WC1H 0AH

Dr. J. Peter Kincaid
Army Research Institute
Orlando Field Unit
c/o PM TRADE-E
Orlando, FL 32813

Dr. Douglas B. Lenat
MCC
9430 Research Blvd.
Echelon Building #2
Austin, TX 78759

Dr. Lewis Johnson
USC
Information Sciences Institute
4670 Admiralty Way, Suite 1001
Marina Del Rey, CA 90292

Dr. Milton S. Katz
European Science Coordination
Office
U.S. Army Research Institute
Box 65
PPO New York 09510-1500

Dr. Walter Kintsch
Department of Psychology
University of Colorado
Boulder, CO 80309-0245

Dr. Alan M. Lesgold
Learning R and D Center
3939 O'Hara Street
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Daniel B. Jones
U.S. Nuclear Regulatory
Commission
NRR/ILRB
Washington, DC 20555

Dr. Smadar T. Kedar-Cabelli
NASA Ames Research Center
Mail Stop 244
Moffett Field, CA 94035

Dr. Yves Kodratoff
George Mason University
AI Center
Fairfax, VA 22030-4444

Dr. Keith R. Levi
Honeywell S and RC
3880 Technology Drive
Minneapolis, MN 55418

Mr. Paul L. Jones
Research Division
Chief of Naval Technical Training
Building Post-1
Naval Air Station Memphis
Millington, TN 38054-5056

Dr. Frank Keil
Department of Psychology
228 Uris Hall
Cornell University
Ithaca, NY 14850

Dr. Janet Kolodner
School of Information and
Computer Science
Georgia Institute of Technology
Atlanta, GA 30332-0280

Dr. John Levine
Learning R and D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Randolph Jones
Information and Computer Science
University of California
Irvine, CA 92717

Dr. Richard M. Keller
Knowledge Systems Laboratory
Stanford University
Computer Science Dept.
Stanford, CA 94305

Dr. Stephen Kosslyn
Harvard University
1238 William James Hall
33 Kirkland St.
Cambridge, MA 02138

Dr. Leon S. Levy
2D Dorado Drive
Morristown, NJ 07960

ONR DISTRIBUTION LIST [ILLINOIS/WILKINS]

Matt Lewis
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Doris K. Lidtke
Software Productivity Consortium
1880 Campus Commons Drive, North
Reston, VA 22091

Dr. Sridhar Mahadevan
Dept. of Computer Science
Rutgers University
New Brunswick, NJ 08903

Vern M. Malec
NPRDC, Code 14
San Diego, CA 92152-6800

Dr. Jane Malin
Mail Code EF5
NASA Johnson Space Center
Houston, TX 77058

Dr. William L. Maloy
Code 04
NETPMSA
Pensacola, FL 32509-5000

Dr. Michel Manago
IntelliSoft
28 rue Georges Clemenceau
91400, ORSAY, FRANCE

Dr. William Mark
Lockheed AI Center
Building 259
2710 Sand Hill Rd.
Department 9006
Menlo Park, CA 94025

Dr. Sandra P. Marshall
Dept. of Psychology
San Diego State University
San Diego, CA 92182

Dr. Manton M. Matthews
Department of Computer Science
University of South Carolina
Columbia, SC 29208

Mr. John Mayer
University of Michigan
702 Church Street
Ann Arbor, MI 48104

Dr. John McDermott
DEC
DIB6-3/E2
290 Donald Lynch Blvd.
Marlboro, MA 01752

Prof. David D. McDonald
Department of Computer
& Information Sciences
University of Massachusetts
Amherst, MA 01003

Dr. Joseph C. McLachlan
Code 51
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Barbara Means
SRI International
331 Ravenswood Avenue
Menlo Park, CA 94025

Dr. Douglas L. Medin
Psychology Department
Perry Building
University of Michigan
330 Packard Rd.
Ann Arbor, MI 48104

Dr. Jose Mestre
Department of Physics
Hasbrouck Laboratory
University of Massachusetts
Amherst, MA 01003

Dr. Theodore Metzler
Department of the Navy
Office of the Chief of Naval Research
Arlington, VA 22217-5000

Dr. Alan L. Meyrowitz
Office of Naval Research, Code 1433
800 N. Quincy Rd.
Arlington, VA 22217

Dr. Ryssard S. Michalski
Department of Computer Science
George Mason University
4400 University Drive
Fairfax, Va 22030

Dr. Donald Michie
The Turing Institute
George House
36 North Hanover Street
Glasgow G1 2AD
UNITED KINGDOM

Dr. Vittorio Midoro
CNR-Istituto Tecnologie Didattiche
Via All'Opera Pia 11
GENOVA-ITALIA 16145

Dr. James R. Miller
MCC
3500 W. Balcones Center Dr.
Austin, TX 78759

Prof. Perry L. Miller
Dept. of Anesthesiology
Yale University School
of Medicine
333 Cedar Street
P. O. Box 3333
New Haven, CT 06510

Dr. Christine M. Mitchell
School of Indus. and Sys. Eng.
Center for Man-Machine
Systems Research
Georgia Institute of Technology
Atlanta, GA 30532-0205

Dr. Tom M. Mitchell
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Dr. Sanjay Mittal
Knowledge System Area
Intelligent Systems Lab
Xerox Palo Alto Research Center
Palo Alto, CA 94304

Dr. Andrew R. Molnar
Applic. of Advanced Technology
Science and Engr. Education
National Science Foundation
Washington, DC 20550

Dr. William Montague
Naval Personnel R and D Center
San Diego, CA 92152-6800

Dr. Melvin D. Montemerlo
NASA Headquarters
Code RC
Washington, DC 20546

Dr. Raymond Mooney
Dept. of Computer Sciences
The University of Texas at Austin
Taylor Hall 2.124
Austin, TX 78712

Dr. Katharina Morik
GMD
P3/XPS
P.O. Box 1240
D-5205 St. Augustin
WEST GERMANY

Prof. John Morton
MRC Cognitive
Development Unit
17 Gordon Street
London WC1H 0AH
UNITED KINGDOM

Dr. Jack Mostow
Dept. of Computer Science
Rutgers University
New Brunswick, NJ 08903

Dr. Randy Mumaw
Training Research Division
HumRRO
1100 S. Washington
Alexandria, VA 22314

Dr. Allen Munro
Behavioral Technology
Laboratories - USC
1845 S. Elena Ave., 4th Floor
Redondo Beach, CA 90277

Dr. Kenneth S. Murray
Dept. of Computer Sciences
University of Texas at Austin
Taylor Hall 2.124
Austin, TX 78712

Dr. William R. Murray
FMC Corporation
Central Engineering Labs
1205 Coleman Avenue
Box 580
Santa Clara, CA 95052

Prof. Makoto Nagao
Dept. of Electrical Engineering
Kyoto University
Yoshida-Honmachi
Sakyo-Ku
Kyoto
JAPAN

Mr. J. Nelissen
Twente University of Technology
Fac. Bibl. Toegepaste Onderwijskunde
P. O. Box 217
7500 AE Enschede
The NETHERLANDS

Dr. T. Niblett
The Turing Institute
George House
36 North Hanover Street
Glasgow G1 2AD
UNITED KINGDOM

Dr. Ephraim Nissan
Department of Mathematics
& Computer Science
New Campus
Ben Gurion University of Negev
P. O. Box 653
84105 Beer-Sheva
ISRAEL

Dr. A. F. Norcio
Code 5530
Naval Research Laboratory
Washington, DC 20375-5000

Dr. Donald A. Norman
C-015
Institute for Cognitive Science
University of California
La Jolla, CA 92093

Dr. Harold F. O'Neil, Jr.
School of Education - WPH 801
Department of Educational
Psychology & Technology
University of Southern California
Los Angeles, CA 90089-0021

Dr. Paul O'Rourke
Department of Information and
Computer Science
University of California
Irvine, CA 92717

Dr. Stellan Ohlsson
Learning R and D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. James B. Olsen
WICAT Systems
1875 South State Street
Orem, UT 84058

ONR DISTRIBUTION LIST [ILLINOIS/WILKINS]

Dr. Judith Reitman Olson
Graduate School of Business
University of Michigan
701 Tappan
Ann Arbor, MI 48109-1234

Admiral Piper
PM TRADE
ATTN: AMCPM-TNO-ET
12350 Research Parkway
Orlando, FL 32826

Dr. Stephen Reder
NWREL
101 SW Main, Suite 600
Portland, OR 97204

Dr. Paul S. Rosenbloom
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, CA 90292

Office of Naval Research,
Code 1142CS
800 N. Quincy Street
Arlington, VA 22217-5000
(6 Copies)

Dr. Peter Pizolli
Graduate School of Education
EMST Division
4533 Tolman Hall
University of California, Berkeley
Berkeley, CA 94702

Dr. James A. Reggia
University of Maryland
School of Medicine
Department of Neurology
22 South Greene Street
Baltimore, MD 21201

Dr. Ernst Z. Rothkopf
AT&T Bell Laboratories
Room 2D-468
600 Mountain Avenue
Murray Hill, NJ 07974

Dr. Judith Orasanu
Basic Research Office
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dept. of Administrative Sciences
Code 54
Naval Postgraduate School
Monterey, CA 93943-5026

Dr. J. Wesley Regan
AFHRL/IDI
Brooks AFB, TX 78235

Dr. Allen A. Rovick
Rush Medical College
1653 W. Congress Parkway
Chicago, IL 60612-3864

Dr. Jesse Orlansky
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311

Dr. Tomaso Poggio
Massachusetts Institute
of Technology E25-201
Center for Biological
Information Processing
Cambridge, MA 02139

Dr. Brian Reiser
Cognitive Science Lab
Princeton University
221 Nassau Street
Princeton, NJ 08544

Dr. Stuart J. Russell
Computer Science Division
University of California
Berkeley, CA 94720

Prof. Tim O'Shea
Institute of Educational Technology
The Open University
Walton Hall
Milton Keynes MK7 6AA
Buckinghamshire, U.K.

Dr. Peter Polson
University of Colorado
Department of Psychology
Boulder, CO 80309-0345

Dr. Lauren Resnick
Learning R & D Center
University of Pittsburgh
3929 O'Hara Street
Pittsburgh, PA 15213

Dr. Roger C. Schank
Northwestern University
Inst. for the Learning Sciences
1890 Maple
Evanston, IL 60208

Dr. Everett Palmer
Mail Stop 239-3
NASA-Ames Research Center
Moffett Field, CA 94035

Dr. Bruce Porter
Computer Science Department
University of Texas
Taylor Hall 2.124
Austin, TX 78712-1188

Dr. J. Jeffrey Richardson
Center for Applied AI
College of Business
University of Colorado
Boulder, CO 80309-0419

Lowell Schuer
Psychological & Quantitative
Foundations
College of Education
University of Iowa
Iowa City, IA 52242

Dr. Okchoon Park
Army Research Institute
PERI-2
5001 Eisenhower Avenue
Alexandria, VA 22333

Mr. Armand E. Frieditis
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

Prof. Christopher K. Riesbeck
Department of Computer Science
Yale University
P. O. Box 2158, Yale Station
New Haven, CT 06520-2158

Dr. Jeffrey C. Schlimmer
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Dr. Ramesh Patil
MIT
Laboratory for Computer Science
Room 418
545 Technology Square
Cambridge, MA 02139

Dr. Joseph Psotka
ATTN: PERI-IC
Army Research Institute
5001 Eisenhower Ave.
Alexandria, VA 22333-5600

Prof. David C. Rine
Department of Computer
& Information Sciences
George Mason University
4400 University Drive
Fairfax, VA 22030

Dr. Janet W. Schofield
818 LRDC Building
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Michael J. Passani
Department of Computer and
Information Science
University of California
Irvine, CA 92717

Dr. J. Ross Quinlan
School of Computing Sciences
N.S.W. Institute of Technology
Broadway
N.S.W. AUSTRALIA 2007

Dr. Edwina L. Rissland
Dept. of Computer and
Information Science
University of Massachusetts
Amherst, MA 01003

Dr. Paul D. Scott
University of Essex
Dept. of Computer Science
Wivenhoe Park
Colchester CO43SQ
ENGLAND

Dr. Roy Pea
Institute for Research
on Learning
2550 Hanover Street
Palo Alto, CA 94304

Dr. Shankar A. Rajamoney
Computer Science Department
University of Southern California
Los Angeles, CA 91030

Dr. Linda G. Roberts
Science, Education, and
Transportation Program
Office of Technology Assessment
Congress of the United States
Washington, DC 20510

Dr. Alberto Segre
Cornell University
Computer Science Department
Upson Hall
Ithaca, NY 14853-7501

Dr. Ray S. Peres
ARI (PERI-II)
5001 Eisenhower Avenue
Alexandria, VA 22333

Mr. Paul S. Rau
Code U-33
Naval Surface Weapons Center
White Oak Laboratory
Silver Spring, MD 20903

LT CDR Michael N. Rodgers
Canadian Forces Personnel
Applied Research Unit
4900 Yonge Street, Suite 600
Willowdale, Ontario M2N 6B7
CANADA

Dr. Colleen M. Seifert
Dept. of Psychology
University of Michigan
350 Packard Rd.
Ann Arbor, MI 48104

Dr. C. Perrino, Chair
Dept. of Psychology
Morgan State University
Cold Spring Ln.-Hillen Rd.
Baltimore, MD 21239

Ms. Margaret Recker
Graduate Group in Education
EMST Division
4533 Tolman Hall
University of California, Berkeley
Berkeley, CA 94702

Dr. Oliver G. Selfridge
GTE Labs
Waltham, MA 02254

ONR DISTRIBUTION LIST [ILLINOIS/WILKINS]

Dr. Michael G. Shafte
NASA Ames Research Ctr.
Mail Stop 239-1
Moffett Field, CA 94035

Dr. Derek Sleeman
Computing Science Department
The University
Aberdeen AB9 2PX
Scotland
UNITED KINGDOM

Dr. Patrick Suppes
Stanford University
Institute for Mathematical
Studies in the Social Sciences
Stanford, CA 94305-4115

Dr. Kurt Van Lehn
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Valerie L. Shalin
Honeywell S&RC
3660 Technology Drive
Minneapolis, MN 55418

Ms. Gail K. Slemen
LOGICON, Inc.
P.O. Box 45168
San Diego, CA 92138-5168

Dr. Richard Sutton
GTE Labs
Waltham, MA 02254

Dr. W. S. Vaughan
800 N. Quincy Street
Arlington, VA 22217

Dr. Jude W. Shavlik
Computer Sciences Department
University of Wisconsin
Madison, WI 53706

Dr. Edward E. Smith
Department of Psychology
University of Michigan
330 Packard Road
Ann Arbor, MI 48103

Dr. William Swartout
USC
Information Sciences Institute
4678 Admiralty Way
Marina Del Rey, CA 90293

Dr. Adrian Walker
IBM
P. O. Box 704
Yorktown Heights, NY 10598

Mr. Colin Sheppard
AXC2 Block 2
Admiralty Research Establishment
Ministry of Defence Portsmouth
Portsmouth Hants PO84AA
UNITED KINGDOM

Dr. Reid G. Smith
Schlumberger Technologies Lab.
Schlumberger Palo Alto Research
2240 Hillview Avenue
Palo Alto, CA 94304

Mr. Prasad Tadepalli
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

Dr. Diana Wearne
Department of Educational
Development
University of Delaware
Newark, DE 19711

Dr. Ben Shneiderman
Dept. of Computer Science
University of Maryland
College Park, MD 20742

Dr. Elliot Soloway
EE/CS Department
University of Michigan
Ann Arbor, MI 48109-2122

Dr. Gheorghe Tecuci
Research Institute for Computers
and Informatics
71218, Bd. Miciurin 8-10
Bucharest 1
ROMANIA

Prof. Shalom M. Weiss
Department of Computer Science
Hill Center for Mathematical
Sciences
Rutgers University
New Brunswick, NY 08903

Dr. Jeff Shrager
Xerox PARC
3333 Coyote Hill Rd.
Palo Alto, CA 94304

Linda B. Sorisio
IBM-Los Angeles Scientific Center
11601 Wilshire Blvd., 4th Floor
Los Angeles, CA 90025

Dr. Perry W. Thorndyke
FMC Corporation
Central Engineering Labs
1205 Coleman Avenue, Box 580
Santa Clara, CA 95052

Dr. Keith T. Wescourt
FMC Corporation
Central Engineering Labs
1205 Coleman Ave., Box 580
Santa Clara, CA 95052

Dr. Howard Shrobe
Symbolics, Inc.
Eleven Cambridge Center
Cambridge, MA 02142

Dr. N. S. Sridharan
FMC Corporation
Box 580
1205 Coleman Avenue
Santa Clara, CA 95052

Dr. Chris Tong
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

Dr. David C. Wilkins
Dept. of Computer Science
University of Illinois
405 N. Mathews Avenue
Urbana, IL 61801

Dr. Randall Shumaker
Naval Research Laboratory
Code 5510
4555 Overlook Avenue, S.W.
Washington, DC 20375-5000

Dr. Frederick Steinheiser
CIA-ORD
Ames Building
Washington, DC 20505

Dr. Douglas Towne
Behavioral Technology Labs
University of Southern California
1845 S. Elena Ave.
Redondo Beach, CA 90277

Dr. Kent E. Williams
Institute for Simulation and Training
The University of Central Florida
12424 Research Parkway, Suite 200
Orlando, FL 32826

Dr. Bernard Siler
Information Sciences
Fundamental Research Laboratory
GTE Laboratories, Inc.
40 Sylvan Road
Waltham, MA 02254

Dr. Ted Steinke
Dept. of Geography
University of South Carolina
Columbia, SC 29208

Lt. Col. Edward Trautman
Naval Training Systems Center
12250 Research Parkway
Orlando, FL 32826

Dr. Marsha R. Williams
Appl. of Advanced Technologies
National Science Foundation
SEE/MDRISE
1800 G Street, N.W., Room 925-A
Washington, DC 20550

Dr. Herbert A. Simon
Departments of Computer Science and
Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Leon Sterling
Dept. of Computer Engineering
and Science
Crawford Hall
Case Western Reserve University
Cleveland, Ohio 44108

Dr. Paul T. Towhig
Army Research Institute
ATTN: PERI-RL
5001 Eisenhower Avenue
Alexandria, VA 22333-5800

S. H. Wilson
Code 5505
Naval Research Laboratory
Washington, DC 20375-5000

Robert L. Simpson, Jr.
DARPA/ISTO
1400 Wilson Blvd.
Arlington, VA 22209-2308

Dr. Michael J. Strait
UMUC Graduate School
College Park, MD 20742

Dr. Paul E. Utgoff
Department of Computer and
Information Science
University of Massachusetts
Amherst, MA 01002

Dr. Patrick H. Winston
MIT Artificial Intelligence Lab.
545 Technology Square
Cambridge, MA 02139

Dr. Zita M. Simutis
Chief, Technologies for Skill
Acquisition and Retention
ARI
5001 Eisenhower Avenue
Alexandria, VA 22232

Dr. Devika Subramanian
Department of Computer Science
Cornell University
Ithaca, NY 14853

Dr. Harold P. Van Cott
Committee on Human Factors
National Academy of Sciences
2101 Constitution Avenue
Washington, DC 20418

Dr. Edward Wisniewski
Honeywell S and RC
3660 Technology Drive
Minneapolis, MN 55418

ONR DISTRIBUTION LIST [ILLINOIS/WILKINS]

Dr. Paul T. Wehig
Army Research Institute
5001 Eisenhower Ave.
ATTN: PERI-RL
Alexandria, VA 22233-6000

Dr. Joseph Wohl
Alphatech, Inc.
2 Burlington Executive Center
111 Middlesex Turnpike
Burlington, MA 01803

Dr. Beverly P. Woolf
Dept. of Computer and
Information Sciences
University of Massachusetts
Amherst, MA 01003

Dr. Ronald R. Yager
Machine Intelligence Institute
Iona College
New Rochelle, NY 10801

Dr. Masoud Yasdani
Dept. of Computer Science
University of Exeter
Prince of Wales Road
Exeter EX44PT
ENGLAND

Dr. Joseph L. Young
National Science Foundation
Room 320
1800 G Street, N.W.
Washington, DC 20550

Dr. Maria Zemankova
National Science Foundation
1800 G Street N.W.
Washington, DC 20550

Dr. Uri Zernik
GE - CRD
P. O. Box 8
Schenectady, NY 12301