(12)

The Interaction Between Knowledge and Practice in the Acquisition of Cognitive Skills

Stellan Ohlsson
The Learning Research and Development Center,
University of Pittsburgh,
Pittsburgh, PA 15260

June, 1992

Final Technical Report
ONR Contract No. N00014-89-J-1681

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704 0188

[illegible]

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
July, 1992

3. REPORT TYPE AND DATES COVERED
Final Tech Report - 3/1/89 - 8/31/92

4. TITLE AND SUBTITLE
The Interaction Between Knowledge and Practice in the Acquisition of Cognitive Skills

5. FUNDING NUMBERS
N00014-89-J-1681

6. AUTHOR(S)
Stellan Ohlsson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Learning Research and Development Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

8. PERFORMING ORGANIZATION REPORT NUMBER
U Pitt/LRDC/
ONR/KUL-92-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Cognitive Science Program
Office of Naval Research (Code 1142C5)
800 North Quincy Street
Arlington, VA 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
44208

11. SUPPLEMENTARY NOTES
To appear in: A. Meyrowitz and S. Chipman, (Eds.), Cognitive Models of Complex Learning. Norwell, MA: Kluwer Academic Publishers. In press.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The role of prior knowledge in skill acquisition is to enable the learner to detect and to correct erros. Computational mechanisms that carry out these two functions are implemented in a simulation model which represents prior knowledge in constraints. The model learns symbolic skills in mathematics and science by noticing and correcting constraint violations. Results from simulation runs include quantitative predictions about the learning curve and about transfer of training. Because constraints can represent instructions as well as prior knowledge, the model also simulates one-on-one tutoring. The implications for the design of instruction include a detailed specification of the content of effective feedback messages for intelligent tutoring systems.

14. SUBJECT TERMS Key words:

15. NUMBER OF PAGES
62

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

To appear in: A. Meyrowitz and S. Chipman, (Eds.), *Cognitive Models of Complex Learning*. Norwell, MA: Kluwer Academic Publishers. In press.

The Interaction Between Knowledge and Practice in the Acquisition of Cognitive Skills*

Stellan Ohlsson
Learning Research and Development Center
University of Pittsburgh

Abstract

The role of prior knowledge in skill acquisition is to enable the learner to detect and to correct errors. Computational mechanisms that carry out these two functions are implemented in a simulation model which represents prior knowledge in *constraints*. The model learns symbolic skills in mathematics and science by noticing and correcting constraint violations. Results from simulation runs include quantitative predictions about the learning curve and about transfer of training. Because constraints can represent instructions as well as prior knowledge, the model also simulates one-on-one tutoring. The implications for the design of instruction include a detailed specification of the content of effective feedback messages for intelligent tutoring systems.

* The research reported in this chapter was supported by Grant No. N00014-89-J-1681 from the Cognitive Science Program of the Office of Naval Research. The opinions expressed are not necessarily those of the sponsoring agency, and no endorsement should be inferred. Approved for public release; distribution unlimited.

THE ROLE OF KNOWLEDGE IN LEARNING

Learning and knowledge are doubly related. On the one hand, knowledge is the outcome of learning. On the other hand, knowledge is one of the inputs into the learning process. New skills are constructed within the context provided by prior knowledge. This is no less true of technical domains such as mathematics, science, and engineering than of common sense domains such as cooking and travel planning.

Cognitive scientists from Ebbinghaus (1964/1885) to VanLehn (1982) have sought to escape the complexities of prior knowledge by studying situations in which such knowledge plays a minimal role. This simplification has payed off theoretically. Following the pioneering papers by Anzai and Simon (1979) and by Anderson, Kline, and Beasley (1979) several computational models of the acquisition of cognitive skills in the absence of prior knowledge have been proposed (e. g., Anderson, 1983; Holland et al., 1986; Langley, 1987; Ohlsson, 1987a; Rosenbloom, 1986; VanLehn, 1990). These models assume that procedural knowledge forms a closed loop: Problem solving methods generate problem solving steps which, in turn, generate the experiences from which new problem solving methods are induced. Simulation models of this kind constitute an important advance over the mathematical and verbal learning theories of the past, but the learning mechanisms proposed within this paradigm (chunking, composition, discrimination, generalization, grammar induction, subgoaling, etc.) do not explain the role of prior knowledge in learning. There is no point along the method-step-method loop at which domain knowledge can impact the learning process.

Empirical research of knowledge-based skill acquisition began with Judd's (1908) study of the skill of throwing darts at underwater targets with and without knowledge of the principle of refraction. Both he and later Katona (1940) reported dramatic effects of knowledge about underlying principles on skill acquisition. Kieras and Bovair (1984) also found such an effect, but other recent studies have found weaker effects or no effect (e. g., Gick & Holyoak, 1983; Smith & Goodman, 1984). Educational researchers frequently report that instruction in the relevant domain knowledge does not guarantee correct action (e. g., Resnick & Omanson, 1987; Reif, 1987). On the other hand, inappropriate prior knowledge--so-called misconceptions--is quite likely to interfere with successful problem solving (Confrey, 1990). The empirical results indicate that we do not yet understand how prior knowledge interacts with skill acquisition well enough to ask the right experimental questions.

Theoretical analysis of the function of prior knowledge in skill acquisition has hardly began. Ohlsson (1987b) proposed a computer model which explained how inferential knowledge about the domain enables a learner to find a more efficient strategy for a task which he or she already knows how to solve. The hypothesis behind this model was that domain knowledge allows the learner to reason about possible simplifications of his or her current strategy. The model simulated speed-up of a simple reasoning strategy, but it threw no light on the role of domain knowledge in the initial acquisition of that strategy.

The purpose of the work reported here is to explore the hypothesis that ***the function of knowledge in initial skill acquisition is to enable the learner to detect and correct errors.*** This hypothesis is embodied in a running simulation model which uses prior knowledge to learn cognitive skills from unguided practice. The theory predicts the negatively accelerated practice curve observed in human learning, throws some new light on the problem of transfer of training, and suggests an analysis of tutoring with some very specific implications for the design of intelligent tutoring systems.

Throughout this chapter, the terms "domain knowledge" and "prior knowledge" refer to declarative knowledge, while the terms "cognitive skill", "problem solving method", "decision rule", and "mental procedures" refer to procedural knowledge. Both common sense and philosophy have long distinguished between theory and practice, between knowing ***that*** and knowing ***how***, but the particular formulation of this distinction used here is imported from Artificial Intelligence (Winograd, 1975).

Procedural knowledge is prescriptive and use-specific. To a first approximation, it consists of associations between goals, situations, and actions. Examples of procedural knowledge are place-value algorithms for arithmetic, methods for electronic trouble shooting, explanatory strategies in biology, and the procedure for constructing structural formulas for organic molecules. Declarative knowledge, on the other hand, is descriptive (as opposed to prescriptive) and use-independent. To a first approximation, it consists of facts and principles. Examples of declarative knowledge are the laws of the number system, the general gas law, Darwin's theory of evolution, and the theory of the co-valent bond. The function of procedural knowledge is to control action; the function of declarative knowledge is to provide generality. Intelligent behavior requires both types of knowledge (Anderson, 1976; Winograd, 1975).

If the two types of knowledge are distinct, how do they interact? In particular, if declarative knowledge is use-independent and distinct from procedures, then how does it influence action? The problem investigated in the research program summarized in this chapter is how (previously learned) declarative knowledge affects the construction of (new) procedural knowledge.

A FUNCTIONAL THEORY OF SKILL ACQUISITION

Learning happens during problem solving; to learn is to adapt to the structure of the task environment; learning is triggered by contradictions between the outcomes of problem solving steps and prior knowledge. These three principles imply a particular functional breakdown of skill acquisition.

Principle 1: Learning as Problem Solving

During practice, the learner is faced with problems which he or she does not yet know how to solve--that is why he or she is practicing. Practice is problem solving and skill acquisition is the encoding of the results of problem solving for future use. People solve unfamiliar problems with so-called weak methods, i. e., problem solving methods which are so general that they can be applied even with a minimum of information about the task environment. The weak methods people have been observed to use include analogical inference, hill climbing, forward search, means-ends analysis, and planning.

Weak methods are general but inefficient. The function of weak methods during practice is not to produce complete or correct problem solutions, but to generate task relevant behavior. Activity vis-a-vis the task provides the learner with the opportunity to discover the structure of the task environment. Cognitive skills are constructed by interpreting, storing, and indexing such discoveries so that they can be retrieved and applied later. The function of weak methods is to provide learning opportunities, not to solve problems.

Individual weak methods were formalized in the late fifties and early sixties (Feigenbaum & Feldman, 1963), but the general category of weak methods was first identified by Newell (1969, 1980). Laird (1986) has suggested that there exists a universal weak method from which all other weak methods can be derived.

The idea that learning is problem solving and that the function of weak methods is to provide learning opportunities is implicit in the concept of trial and error and thus traces its roots back to behaviorism. Although first formalized in a computational model by Anzai and Simon (1979), this idea is central to several recent models of learning (e. g., Anderson, 1986; Holland et al., 1986; Rosenbloom, 1986). In the field of machine learning, the notion that learning occurs *en route* to an answer rather than after completion of a practice problem has been emphasized by Mostow and Bhatnager (1987, 1990) in their work on adaptive search.

Principle 2: Learning as Adaptation

Weak methods are inefficient because they are general. A domain-specific cognitive skill is efficient because it reflects the structure of the relevant task environment. Skill acquisition begins with maximally general procedures (weak methods) and ends with domain-specific skills. Learning is gradual adaptation.

The process of adaptation cannot continue indefinitely. The task environment only contains so much structure and when all the structure has been absorbed, the skill cannot get any more specific or better adapted. In complex and irregular domains, expert strategies are between weak methods and algorithms in specificity. They guide behavior without fully determining it and considerable uncertainty can remain even at the highest level of expertise.

The idea that learning proceeds from the general to the specific is counterintuitive, because it is common sense that learning begins with the concrete and the specific and moves towards the general. The common sense theory has little support in systematic research. Formal analyses of induction (e. g., Angluin & Smith, 1983) have revealed that many induction problems are NP-complete and that noisy input cripples most induction algorithms. David Hume was right; induction does not work. Knowledge must be constructed in some other way. Specialization of pre-existing, general structures is one alternative. The particular version of this idea in which learning proceeds from general *methods* to task-specific *methods* was implicit in early computational models (e. g., Anzai & Simon, 1979), but was to the best of my knowledge first stated in two papers by Langley (1985) and by Anderson (1987).

The idea that learning is adaptation to the environment can be formulated in many different ways, as a comparison between Hull (1943), Piaget (1971), and Anderson (1990) demonstrates. Until recently, psychologists lacked a formal method for describing the learner's environment independently of the learner. This threatened to make the principle of adaptation circular, or at least difficult to apply. The information processing approach is a major breakthrough because it provides a formal description of task environments. Specifically, an environment is described as a *search space* (or problem space; Newell & Simon, 1972). The organism is then naturally described as a strategy for traversing that space. Adaptation has a very definite meaning within this formalization: A given strategy is adapted to a particular task environment in inverse proportion to the amount of search required by that strategy to find a path from the initial state to the goal state. A maximally adapted strategy is one which leads to the goal without extra or unnecessary steps.[1]

---

[1]In an alternative approach, Anderson (1990) describes the environment in terms of its statistical regularities. Many memory phenomena follow from the assumption

Principle 3: Learning as Conflict Resolution

Novices make many errors; that is why we call them novices. Experts do not; that is why we call them experts. The weak methods employed by novices produce errors because they are overly general, causing problem solving steps to be performed in situations in which they are not appropriate. The task-specific skills of experts do not generate errors because they constrain actions to situations in which they are appropriate. The process of adapting a general method to a particular task environment is a process of gradually eliminating errors. Error elimination consists of two subprocesses: error detection and error correction.

Error Detection. Learners can detect their errors in three ways: by observing environmental effects, by self-monitoring, and by being told by others (Reason, 1990, Chap. 6). Some task environments provide direct feedback about errors. If the unknown device exploded when the red button was pushed, pushing the red button was an error. Other task environments do not provide feedback of this sort. In such environments, learners can detect their errors by checking new conclusions against their prior knowledge. Incomplete or incorrect procedural knowledge is highly likely to generate conclusions or problem states that contradict what the learner knows is true of the domain.

As an illustration, consider the following everyday situation: You are driving to an unfamiliar location with the instruction to follow route X north and make a right-hand turn onto Y-street. You are looking for the turn and not finding it. Did you overshoot the turn or did you not go far enough? The only way to decide whether you missed your turn is to know some landmark (e. g., a bridge) which is further out on route X than the turn onto Y-street. (A thoughtful friend includes such a landmark in his or her instructions.) When you see the landmark, you know that you missed your turn. The contradiction between the prior knowl-

---

that memory is adapted to those regularities (Anderson & Schooler, 1991). Anderson (1993) applies this approach to skill acquisition as well.

edge that "Y-street is before the bridge" and the observation "here is the bridge now" allows you to recognize that you have made a mistake.

Technical skills often apply in symbolic task environments in which contradictions between outcomes of problem solving steps and prior knowledge constitute the only indicators of errors. Mathematical symbols do not complain about being inserted into false equalities, unsolvable equations, or incorrect calculations, so a good learner checks his or her calculations. Checking, say, a subtraction by adding the difference and the subtrahend requires the knowledge that the sum of the difference and the subtrahend ought to equal the minuend. Structural formulas for organic molecules do not beep when the laws of the co-valent bond are violated. Noticing an error in a structural formula requires the knowledge that each bond ought to be associated with exactly two electrons, that the total number of electrons cannot exceed the number of valence electrons for the molecule, and so on. The more knowledge, the higher the probability that the learner can detect his or her errors.

Error Correction. The detection of a contradiction between a new conclusion and prior knowledge leads to processes that aim to restore consistency by revising the relevant procedural knowledge. If the execution of action A in situation S_1 leads to a new situation S_2 which violates some principle of the domain, then the mental decision procedure that chose A in S_1 is faulty. The obvious correction is to constrain the procedure so as to avoid executing A in situations like S_1. This requires that the learner identifies the conditions that caused the error, i. e., those properties of S_1 that guaranteed that the error would occur if A were executed. Given knowledge of those conditions, the mental procedure can be revised so as to avoid similar errors in the future.

The principle that learning is error correction superficially resembles Thorndyke's Law of Effect which says that actions with negative consequences are gradually removed from the learner's behavioral repertoire (while actions with positive consequences are strengthened). However, the two principles are distinct, because a cognitive conflict is not necessarily associated with a painful or unpleasant outcome, as the examples given previously illustrate. The error correction principle is also superficially related to the hypothesis that learning is driven by impasses, i. e., situations in which existing procedural knowledge is insufficient to decide what to do next (Newell, 1990; VanLehn, 1988). However, impasses are not errors. An impasse is a situation in which there is insufficient information to make a choice, while an error is a bad choice.

The idea that cognitive change is triggered by contradictions and inconsistencies has been suggested repeatedly in the cognitive sciences. It is central to several recent cognitive models of learning. Holland et al. (1986) put prediction-based evaluation of knowledge at the center of learning: Knowledge is continuously applied in predicting events and rules that lead to wrong predictions are modified. Schank (1982, 1986) has proposed the similar idea that learning is triggered by expectation failures. In developmental psychology, Piaget (1985) designated cognitive conflict, which he called disequilibrium, as the driving force of cognitive development. Empirical investigations support this hypothesis (Murray, Ames, & Botvin, 1977). Social psychologists like Festinger (1957) have proposed that cognitive dissonance causes individuals to revise their beliefs in order to restore consistency (see Abelson et al., 1968, for an overview of cognitive consistency theory). The hypothesis that belief revision serves to maintain consistency has also been proposed by philosophers (Quine & Ullian, 1978) and by science educators (Hewson & Hewson, 1984; Posner et al., 1982).

Machine learning researchers have build systems that learn by resolving conflicts (Hall, 1988; Kocabas, 1991; Rose & Langley, 1986) and by explaining errors (Minton, 1988). The problem of what constitutes a rational response to a contradiction has been studied in logic and Artificial Intelligence under the rubric non-monotonic logic (Gardenfors, 1988; McDermott & Doyle, 1980). Finally, the idea that theory development in science is driven by contradictions between theory and data have been formulated in different ways by Duhem (1991/1914), Kuhn (1970), and Popper (1972/1935). The relevance of these philosophers for psychology is highlighted by Berkson and Wettersten's (1984) attempt to recast Popper's philosophy as a learning theory. In short, the idea of cognitive change as a response to conflict, contradiction, or inconsistency has been proposed by so many researchers independently of each other and in so many different fields that it deserves to be recognized as one of the great unifying principles of the cognitive sciences.

Summary

During practice the learner continuously monitors his or her progress by comparing the current state of the practice problem to his or her prior knowledge about the domain. A problem state that contradicts something that is known to be true of the domain indicates that an error has been made. When such a contradiction is noticed, the current problem solving method is constrained so as to avoid making similar errors in the future. As practice progresses, the general method becomes more and more constrained and better and better adapted to the task environment. Eventually it has become transformed into the correct domain-specific skill and ceases to generate errors.

According to this theory, prior knowledge impacts skill acquisition in two ways. First, knowledge allows the learner to detect his or her errors. Facts and principles of the domain generate implications that an incomplete or incorrect skill is likely to violate or contradict. The more knowledge the learner has, the higher the probability that he or she will be aware of the contradictions and conflicts generated by a faulty solution or a mistaken problem solving step.

Second, prior knowledge allows the learner to identify the conditions that caused the error. Finding the cause of an error might require complicated reasoning about the domain. The more knowledge the learner has, the higher the probability that he or she accurately identifies the cause, which in turn is a prerequisite for successful error correction.

In short, the theory put forth here claims that the function of acquiring new skills through practice consists of three main subfunctions--to generate task-relevant behavior, to identify errors, and to correct errors--each of which, in turn, can be analyzed into subfunctions. The functional analysis is summarized in Figure 1. Although the theory supports qualitative arguments and explanations, the derivation of quantitative behavioral predictions requires a working information processing system.

I. Learn to do unfamiliar task

A. Generate task-relevant actions

1. Apply forward search
   a. Retrieve possible actions
   b. Select action
   c. Execute action

B. Learn from erroneous actions

1. Detect errors
   a. Check consistency between current problem state and prior knowledge after each action

2. Correct error

a. Extract information from error
   i. Identify the conditions under which a particular action is incorrect

b. Revise current task procedure
   i. Constrain procedure so as to avoid that action under those conditions

Figure 1. The functional analysis of learning from error.

A COMPUTATIONAL MODEL

To move from a functional theory to a working model one must specify particular representations and processes that can compute the functions described in the theory. In particular, an implementation of the present theory requires (a) a performance mechanism, including a representation for procedural knowledge, (b) a representation for declarative knowledge, (c) a mechanism for detecting errors, and (d) a mechanism for correcting errors. The particular model described here is called the *Heuristic Searcher* (HS).

A Standard Performance Mechanism

Memory Architecture. HS has three memory stores. The *working memory* holds the model's knowledge state, corresponding to the learner's perception of the current state of the practice problem. The *procedural memory* holds the model's procedural knowledge, corresponding to the learner's previously acquired skills. The *long-term memory* holds the model's declarative knowledge, corresponding to the learner's prior knowledge about the domain. There is no separate goal stack. Goals are represented in working memory.

Procedural Knowledge. Procedural knowledge is represented in so-called production rules (Newell & Simon, 1972), i. e., rules of the general form

Goal, Situation --> Action,

where *Goal* is a description of what the learner believes he or she is supposed to achieve in the practice problem, e. g., "construct the structural formula for C_2H_5OH," and *Situation* is a description of a class of situations, e. g., "situations in which the carbon skeleton of the molecule has been completed but no other atoms have been connected yet." Formally speaking, both *Goal* and *Situation* are patterns, i. e., conjunctions of elementary propositions which may or may not contain (universally quantified) variables.

The action on the right-hand side of a production rule is a problem solving step that the model knows how to perform, e. g., "connect the oxygen atom to one of the carbon atoms". Actions have applicability conditions that have to be satisfied before they can be applied. For example, an oxygen atom cannot be attached to a carbon atom unless there is a carbon atom for it to be attached to. Each action is implemented as a piece of Lisp code that revises the current problem state by deleting some propositions and adding others. Syntactically, the actions are so-called Strips operators (Fikes & Nilsson, 1971). Psychologically, the actions correspond to components of the practice problem which are unproblematic for the learner.

Each production rule is a single unit of procedural knowledge, corresponding to a single problem solving heuristic. The skill required to solve problems of a particular type, e. g., to construct structural formulas in chemistry, consists of a collection of interrelated rules. All production rules are stored in the single production memory, without structural divisions between different skills.

Operating Cycle. The model solves problems by searching a problem space. The content of the working memory at the time the system is initialized is the initial state of the search space. The top goal implicitly specifies the goal state. The ensemble of operators consists of the set of actions the model has been given as input. In each cycle of operation, the *Goals* and *Situations* of the rules are matched against the working memory with a version of the RETE pattern matching algorithm developed by Forgy (1982). If a rule matches, its action is executed.

If more than one rule matches the current state, each matching rule is evoked and one new descendant of the current state is generated for each evoked rule. The entire search tree is saved in memory. Each cycle begins with the selection of which search state to install as the current state for that cycle. In some applications of HS, the selection of the current state is based on a task specific evaluation function, in which case the model performs best-first search. If the evaluation function has the right properties and, in addition, the system checks for repeated occurrences of the same state[2], then the model executes the A* algorithm (Pearl, 1984, p. 64). In the absence of any evaluation function, the state to expand next is selected randomly among the immediate descendants of the current state, in which case the model performs depth-first search. In psychological terms, the performance mechanism correspond to the hypothesis that people respond to uncertainty by thinking through alternative actions before deciding what to do next.

A Representation for Declarative Knowledge

The function of procedural knowledge is to control action. The function of declarative knowledge is not equally obvious. Philosophical discussions often assume that the function of declarative knowledge is to provide descriptions of the world ("the cat is on the mat"), predictions about future events ("the sun will rise tomorrow"), or explanations ("it is snowing, because the temperature fell"). The epistemological, logical, and semantic riddles associated with these functions have exercised thinkers in a variety of disciplines for centuries.

The HS model is based on a different view of the nature and function of declarative knowledge. Declarative knowledge is not used either to describe, predict, or explain but *to circumscribe a set of states of the world*. The unit of declarative knowledge is a *constraint*. Constraints can be interpreted descriptively, i. e., as circumscribing the set of possible states of the world. For example, the law of conservation of mass claims that the mass of the reactants in a chemical experiment is equal to the mass of the reaction products. Mass is neither created nor destroyed in a chemical reaction, so the mass of the inputs is always equal to the mass of the outputs. The point of the mass conservation law is that it circumscribes situations in which mass is conserved, which are possible, and separates them from situations in which mass is not conserved and that it rules out the latter as impossible. Figure 2 shows the constraint interpretation of the mass conservation law.

Constraints are not limited to representing abstract principles like the law of conservation. Particular facts are also constraints. For exam-

[2]This facility is computationally expensive and is usually switched off.

Example 1: A scientific principle

Idiomatic English:	Energy cannot be created or destroyed.
Constraint formulation:	If the mass of the reactants for a chemical experiment is M_1 and the mass of the products is M_2, then M_1 must be equal to M_2.
Formal representation:	(Reactants R) (Mass R M_1) (Products P) (Mass P M_2) ** (Equal M_1 M_2)

Figure 2. Encoding a scientific principle as a constraint.

ple, the fact that alcohol molecules have an OH-group corresponds to the constraint that a structural formula for an alcohol had better have an OH-group somewhere. Figure 3 shows the constraint interpretation of this fact.

Constraints can also be interpreted prescriptively, i. e., as circumscribing the set of ***desired*** states of the world. The ordinance that one should not drive along a one-way street in the wrong direction is a constraint. Specifically, the fact that Fifth Avenue is one-way in the westerly direction corresponds to the constraint that if you are driving on Fifth Avenue, you had better be heading west. It is not impossible to head east, it is merely undesirable. Figure 4 shows the constraint interpretation of this ordinance.

It is a mistake to try to classify individual constraints as either descriptive or prescriptive. All constraints can be interpreted in both ways, because the two interpretations determine each other. It is desirable that a chemistry experiment satisfies the constraint that the mass of the reac-

Example 2: A scientific fact

Idiomatic English:	Every alcohol molecule has an OH-group.
Constraint formulation:	If X is an alcohol molecule, then it must have an OH-group.
Formal representation:	(Isa X molecule) (Substance X ALCOHOL) ** (Isa Y OH-GROUP) (Part-of Y X)

Figure 3. Encoding a scientific fact as a constraint.

tants is equal to the mass of the reaction products. If this is not the case, then some error was committed in the execution of the laboratory procedure, i. e., some mass was accidentally lost or the experiment was contaminated in some way (Gensler, 1987). The constraint expressed in the mass conservation law acquires a prescriptive function because it can be interpreted descriptively; a laboratory procedure ought to conform to it precisely because it is true. The descriptive and prescriptive aspects of constraints are inseparable.

The main contribution of the HS model is a formal representation for constraints and a set of processes for using them. A constraint C is represented as an ordered pair

$$\langle C_r, C_s \rangle$$

where C_r is a ***relevance criterion***, i. e., a specification of the circumstances under which the constraint applies, and C_s is a ***satisfaction criterion***, i. e., a condition that has to be met for the constraint to be satis-

Example 3: An everyday fact

Idiomatic English:	Fifth Avenue is a one-way street heading west.
Constraint formulation:	If someone is driving on Fifth Avenue, then he or she ought to travel westwards.
Formal representation:	(State X DRIVING) (Location X FIFTH-AVENUE) ** (Direction X WEST)

Figure 4. Encoding an everday fact as a constraint.

fied. To continue the traffic example, if Fifth Avenue is one-way in the westerly direction, then "driving on Fifth Avenue" is the relevance criterion and "is heading west" is the satisfaction criterion. If I am not on Fifth Avenue, the direction of my travel is not constrained by this ordinance, but when I am on Fifth, then I had better be driving west rather than east. In the mass conservation example, "M_1 is the mass before the reaction and M_2 is the mass after the reaction" is the relevance criterion, while the equality "$M_1 = M_2$" is the satisfaction criterion.

The double star connective (**) that appears in Figures 2-4 is not a symbol for logical implication. Constraints are not inference rules; they do not generate conclusions. Nor are they production rules; they do not fire operators. The semantics of the double star connective is similar to the meaning of "ought to", "had better", and related phrases. The interpretation of a constraint $<C_r, C_s>$ is that whenever C_r is the case, C_s ***ought to*** be the case as well (or else something has gone awry). Syntactically, both C_r and C_s are patterns, i. e., conjunctions of propositions similar to the condition side of a production rule.

The HS model does not have any mechanism for acquiring or revising its declarative knowledge. The constraints are input by the user and they stay unchanged throughout a simulation run. The purpose of the constraints is to facilitate the detection and correction of errors.

A Mechanism for Error Detection

At the beginning of each operating cycle, all production rules are matched against working memory, the rules with matching condition sides are evoked, the actions of those rules are executed, and new problem states thus generated. Each new state is matched against all the available constraints. (The match is computed with the same pattern matcher which matches the production rules.) Constraints with non-matching relevance patterns do not warrant any action on the part of the system, because they are irrelevant. Constraints which have matching relevance patterns and also matching satisfaction patterns are ignored as well. The new state is consistent with the those constraints so no action is required. On the other hand, if a constraint with a matching relevance pattern has a non-matching satisfaction pattern, then the new state violates that constraint and some response or action is called for. Such a *constraint violation* signals that something is wrong with the procedure that generated the current state; an error has been committed.

Specifically, consider a rule R with goal G and a conjunction S of situation features in its left-hand side and a single action A in its right-hand side,

$$R: \; G, S \dashrightarrow A,$$

and a constraint C with relevance pattern C_r and satisfaction pattern C_s,

$$C = \langle C_r, C_s \rangle,$$

where both C_r and C_s are conjunctions of situation features. In particular, let us assume that C_r and C_s each consists of two features:

$$C_r = C_r' \& C_r''$$

and

$$C_s = C_s' \& C_s''.$$

Finally, let us assume that the effect of action A is to add the conjunction of C_r'' and C_s' to the current problem state, i. e.,

$$A = \mathrm{Add}[C_r'' \& C_s'].$$

If a learner with rule R and constraint C encounters a problem state S_1 described by

$$S \& C_r',$$

then the left-hand side of R is satisfied because S is present, so the rule will be evoked and action A executed. The effect is that C_r'' and C_s' are added to S_1, yielding a new problem state S_2 described by

$$S \& C_r' \& C_r'' \& C_s'.$$

In this problem state, both C_r' and C_r'' are present, so C_r matches, i. e., the constraint is relevant. Although C_s' is present, C_s'' is not, so C_s is violated; hence, doing A in situation S_1 was an error.

In principle, there are two possible interpretations of the constraint violation: The fault might lie either with the procedural knowledge--the rule--or with the declarative knowledge--the constraint. Because HS was designed to model skill acquisition, as opposed to the acquisition of declarative knowledge, it assumes that the rule rather than the constraint is at fault.

A Mechanism for Error Correction

A constraint violation is a signal that the procedural knowledge that generated the current problem state is faulty and needs to be revised. HS assumes that the fault lies with the last rule to fire. The problem of how to learn from the constraint violation can be stated as follows: Given that rule R,

$$R: G, S \dashrightarrow A,$$

was applied to state S_1 and that it generated state S_2 and that S_2 violates constraint C, how should the rule be revised? The purpose of the revision is to avoid similar constraint violations in the future. The learning mechanism in the HS model accomplishes this by finding the cause of the constraint violation, i. e., the properties of state S_1 that were responsible for the error, and revising rule R so that it does not apply under those conditions. The learning mechanism finds the relevant properties of S_1 by regressing the violated constraint through the rule with a variant of the standard regression algorithm used in many A. I. systems (Nilsson, 1980, p. 288).

More specifically, rule R is replaced with two new rules R' and R", representing two different revisions of R. The purpose of the first revision is to constrain R so that the new rule will apply only in situations in which constraint C is guaranteed to remain irrelevant. This is accomplished by regressing the relevance pattern through the rule. Continuing the example from the previous subsection, regressing the relevance pattern $C_r = (C_r' \& C_r'')$ through the operator $A = Add[C_r'' \& C_s']$ yields C_r' as the only output (see Nilsson, 1980, p. 288, for an explanation of the regression algorithm). The first new rule is constructed by adding the negation of the output from the regression to the original rule:

$$R': S \;\&\; \textit{not}\; C_r' \dashrightarrow A$$

This rule applies only in those situations in which the constraint is guaranteed to remain irrelevant if action A is executed. Psychologically, the rule corresponds to the knowledge that one should only do A when S is true but C_r' is false (e. g., "if the device needs repair and the power is not on, then open the front panel").

The purpose of the second revision is to constrain rule R so that it applies only in situations in which the constraint C is guaranteed to become both relevant and satisfied if A is executed. This is accomplished by regressing the entire constraint through the rule, instead of the relevance pattern. Regressing (C_r' & C_r'' & C_s' & C_s'') through the operator A = Add[C_r'' & C_s'] yields (C_r' & C_s'') as the output (see Nilsson, 1980, p. 288). The second new rule is constructed by adding this result to the original rule (without negating it):

$$R'': S \mathbin{\&} C_r' \mathbin{\&} C_s'' \dashrightarrow A$$

This rule applies only in those situations in which the constraint is guaranteed to become satisfied if A is executed. Psychologically, the rule corresponds to the knowledge that one should only do A when S, C_r', and C_s'' are all true (e. g., "if the device needs repair, the power is on, and the red light is blinking, then switch off the power").

Figure 5 provides a graphical interpretation of the learning mechanism. The set S of situations in which the original rule R applies is split into three subsets when the rule is revised. The first subset contains those situations in which the constraint is guaranteed to remain irrelevant if action A is executed. They are covered by the first new rule. The second subset contains those situations in which the constraint is guaranteed to become satisfied if A is executed. They are covered by the second new rule. The third subset contains those situations in which doing A leads to a constraint violation. They are thrown away, as it were. Neither of the two new rules apply in those situations, so the error type represented by the third subset has been eliminated.

The fact that one type of error has been eliminated does not imply that the two new rules R' and R" are correct. Although the new rules have been revised so as to be consistent with one constraint, they might

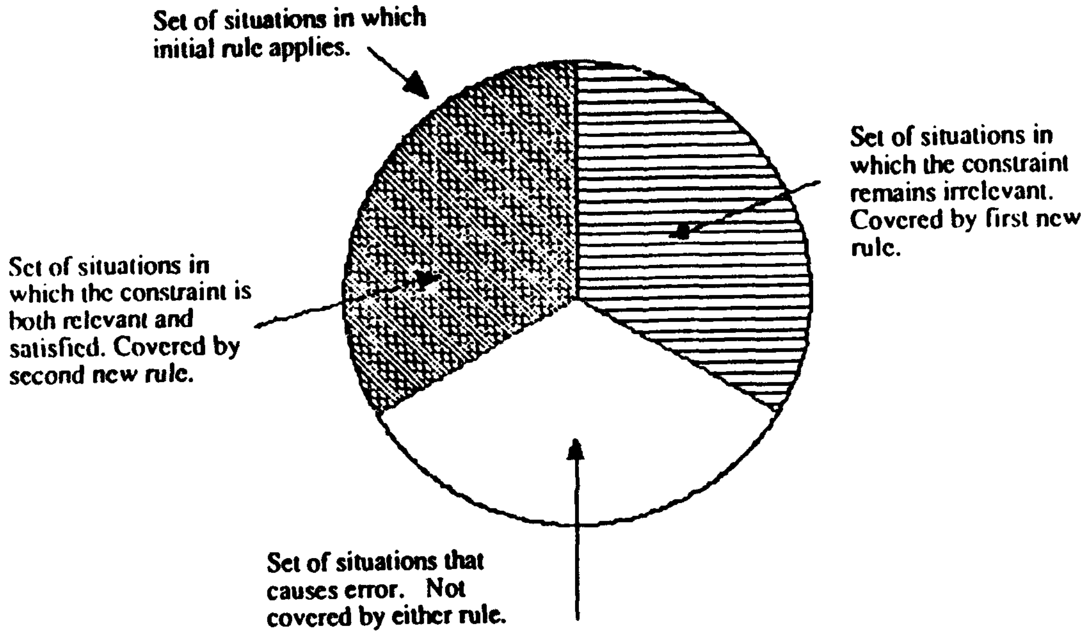

Figure 5. Graphical interpretation of the HS learning mechanism.

still violate other constraints and so have to be revised further. Repeated revisions of rules is the standard case in HS learning. Also, the fact that one rule has been revised does not imply that other rules are correct. Learning proceeds by gradual correction of the relevant rule set as a function of the errors that the model encounters during practice. A detailed analysis of the correction of an entire rule set is available in Ohlsson and Rees (1991a, Table 5).

Discussion

The HS model is based on two representational assumptions: that procedural knowledge is represented in production rules and that declarative knowledge is represented in constraints. The production system format was proposed by Newell and Simon (1972) but has been taken up by other researchers (Klahr, Langley, & Neches, 1987). The main claim of the production system hypothesis is that human action is determined by an external context, represented by the situation the learner is faced with, and an internal context, represented by the learner's goal. Procedural knowledge consists of associations between goals, situations, and actions. The individual production rule is the smallest unit of procedural knowledge; it maps a single goal/situation pair onto a particular action.

A second claim of the production system hypothesis is that the units of procedural knowledge are modular. Production rules do not access or operate upon each other. They only interact through their effects on working memory. There is strong empirical evidence for the modularity of procedural knowledge (Anderson, 1993).

The constraint format originated with the current theoretical effort (Ohlsson & Rees, 1991a) and it does not have any empirical or theoretical support other than the success of the model it is embedded in. There has been so little progress on the epistemological, logical, and semantic problems associated with the standard, propositional interpretation of declarative knowledge that any alternative conception is worth exploring.

Given the two representational assumptions, information processing mechanisms that compute the functions specified in the abstract theory (see Figure 1) can be specified. In HS, the function of generating task relevant activity is carried out by forward search, the function of detecting errors is carried out by a pattern matcher, and the function of correcting errors is carried out by a rule revision algorithm based on regression. There are alternative ways to compute each of these functions. HS could have been implemented with, for example, analogical transfer instead of heuristic search as the weak method responsible for generating task relevant behavior. Similar substitutions of alternative mechanisms are possible for each of the other functions specified in the theory. The predictions generated by running the model are consequences of both the theoretical principles that guided its design and the particular representations and processes that are implemented in it.

Compared to many other machine learning systems, HS is very simple. It combines a standard production system architecture, a well-known weak method, and an off-the-shelf regression algorithm; little else is needed. HS is implemented in Lucid Common Lisp and runs on a Sun Sparcstation 1+ with 16 megabytes of main memory. The core mechanisms have been debugged in hundreds of simulation runs in different domains over a period of four years and are very robust.

APPLICATIONS TO CLASSICAL RESEARCH PROBLEMS

A good theory should throw new light on the perennial problems of the discipline. The learning curve and transfer of training have been central problems in the theory of learning for a long time.

The Learning Curve

Background. If performance level, measured in terms of time to complete a practice problem, is plotted as a function of amount of practice, measured in terms of the number of practice problems solved, i. e., the number of trials, the result is a negatively accelerated curve. The rate of improvement is fastest at the beginning of practice and quickly slows down as mastery is approached. This type of learning curve has been observed in a large number of studies, across many different tasks, and in widely varying subject populations (Lane, 1987; Mazur & Hastie, 1978; Newell & Rosenbloom, 1981; Ohlsson, 1992c).

Armchair reasoning would lead one to expect learning to be slow in the beginning, when the learner is still groping to understand the practice task and there is little relevant knowledge or skill to build on. Later in the practice sequence, the partial knowledge built up during previous trials serves as a lever for acquiring more knowledge, with increased speed of learning as a result. However, research leaves no doubt that the opposite is the case: The rate of skill acquisition is faster the less the learner knows about the task. No theory of practice is viable unless it can explain this unexpected finding.

The hypothesis that skill acquisition is the elimination of errors provides such an explanation. According to this hypothesis, knowledge is revised when the learner becomes aware of an error. Learning is thus a sequence of ***learning events***, with one error (type) being eliminated per event. The prediction of a negatively accelerated learning curve follows from this hypothesis in three easy steps:

1. The consequence of an error is floundering, i. e., unnecessary search. Performance improves when the error is corrected because the unnecessary search is eliminated. Let us assume that the amount of unnecessary search caused by an error is approximately constant across errors. Performance then improves with a constant amount ***per learning event.***
2. At the outset the learner makes many errors on each practice problem precisely because he or she knows so little about the task. As mastery is approached, the number of mistakes per problem decreases because many errors have already been eliminated. There are fewer and fewer learning events ***per trial*** as practice progresses.
3. Constant improvement per learning event and decreasing number of learning events per trial imply a decreasing rate of improvement per trial.

This explanation does not depend on the details of particular information mechanisms. Any theory or model which claims that learning events are triggered by trouble situations--defined as cognitive conflicts, contradictions, errors, expectation failures, impasses, wrong answers or in any other way--implies this explanation, because trouble situations disappear as mastery is approached, by definition of "mastery."

The qualitative argument explains why we should expect the rate of improvement to slow down across trials, but it does not make a specific prediction about the shape of the learning curve. Newell and Rosenbloom (1981) have reviewed the evidence that the human learning curve is a member of the class of curves described by so-called power laws, i. e., by equations of the general form

$$T = A + kP^{-r} \quad (1)$$

where T is the time to complete the current practice problem, A is the asymptotic performance, P is the amount of practice in trials, and k and r are constants.

Simulating the Learning Curve. To derive the learning curve predicted by the present theory, a simulation experiment was run with the HS model. A problem solving skill from the domain of chemistry was chosen as the target for the simulation. Chemists frequently need to know the interconnections between the atoms in a molecule. The interconnections are specified in structural formulas, so-called *Lewis structures*. A Lewis structure shows which atoms in a molecule are bound to which other atoms and by which kind of bond. The task of constructing the Lewis structure for a particular molecule, specified through its molecular (sum) formula, will here be called a *Lewis problem*. Figure 6 shows the initial state and the goal state of a Lewis problem. There is usually more than one path to the goal state. Figure 7 shows one such path. The cognitive skill of solving Lewis problems is taught in the beginning of college level courses in organic chemistry (e. g., Solomons, 1988).

The HS model was given a representation for atoms, molecules, valencies, bonds between atoms, and the other entities, properties and relations that are important in the chemistry environment. The actions involved in Lewis problems are to select atoms, to connect atoms, to

Initial state:

A sum formula

CH_3CH_2OH

Goal state:

A Lewis structure

```
    H   H
    |   |
H - C - C - O - H
    |   |
    H   H
```

Figure 6. Initial state and goal state for a Lewis problem.

make double bonds, and so on. Figure 8 summarizes the problem space for Lewis problems.

In order to attempt to solve practice problems, HS must be given an initial procedure. In this application, the model was given a set of very general initial rules that encode a procedure for how to construct Lewis structures that approximates the verbal recipes given in chemistry textbooks (e. g., Solomons, 1988, pp. 10-11; Sorum & Boikess, 1981, pp. 104-107). Finally, in order to detect and correct its errors, the model must have some prior knowledge about the domain. It was given a set of constraints that encode some relevant facts about the chemistry of alcohols, ethers, and pure hydrocarbons.

Nine molecules--three alcohols, three ethers, and three hydrocarbons--were selected as practice problems. The model solved each of the nine problems, presented in random order. This corresponds to the simulation of a single subject going through a sequence of nine different

1. Connect the carbons: C - C

2. Attach the oxygen: C - C - O

3. Complete the OH-group: C - C - O - H

4. Distribute the hydrogens:

```
    H   H
    |   |
H - C - C - O - H
    |   |
    H   H
```

5. Add electron pairs:

```
    H   H
    |   |
H - C - C - O - H
    |   |
    H   H
```

Figure 7. A solution path for the Lewis problem in Figure 6.

Representation
Symbols that represent atoms, electron pairs, molecules, noble gas configurations, numbers, single, double and tripple bonds, substances, types of carbon arrangements (branched structures, chains, and rings), two-dimensional spatial relations, and valencies.

Initial state
A molecular (sum) formula.

Operators
Select an atom, place the first atom, attach an atom to the molecule, identify open bonds, create multiple bonds, and add electron pairs.

Goal state
A correct Lewis structure for the given molecule. A Lewis structure must (a) connect all the atoms in the sum formula, (b) not include any other atoms than those in the sum formula, (c) have a number of valence electrons equal to the sum of the valence electrons of the atoms, and (d) give each atom a noble gas configuration.

Figure 8. A problem space for Lewis problems.

practice problems. The model was then re-initialized and run through the nine problems once again, simulating a second subject. All in all, the model worked through the nine practice problems 357 times, each time in a different random order, thus simulating a learning experiment with that number of subjects. Figure 9 summarizes the initial knowledge, the training procedure, and the outcome of the chemistry simulation.

The data from the simulation runs were aggregated by averaging the performance of all 357 simulated subjects for each trial. The average performance on each trial was plotted as a function of trial number. (This corresponds to how learning curves are constructed from psycho-

Prior procedural knowledge
The model began with a procedure that connects the heavy atoms, adding multiple bonds if needed, connects the hydrogens, and then adds the final electron pairs. This procedure generates correct Lewis structures, but requires large amounts of search.

Prior declarative knowledge
There were 16 constraints which encode knowledge about (a) properties of particular classes of molecules, e. g., that alcohols have a C-O-H group and that ethers have a C-O-C group, (b) spatial properties of the possible carbon skeletons (branched structures, chains, and rings), and (c) the distribution of hydrogens across the molecule.

Training
The model was given unsupervised practice on a mixed set of Lewis problems that included alcohols, ethers, and pure hydrocarbons.

Learning outcome
The model learned a set of rules for constructing Lewis structures for the relevant molecules with a minimal amount of search.

Figure 9. Summary of the chemistry simulation.

logical data.) Figure 10 shows the results. Performance as a function of practice approximates a straight line when plotted with logarithmic coordinates on both axes, the hallmark of a curve described by a power law. The HS model thus predicts that improvement over time follows the particular shape that has been observed in data from human learning.

The qualitative argument for why learning from error predicts a negatively accelerated learning curve is based on the simplifying assumption that there is a constant improvement per learning event. How

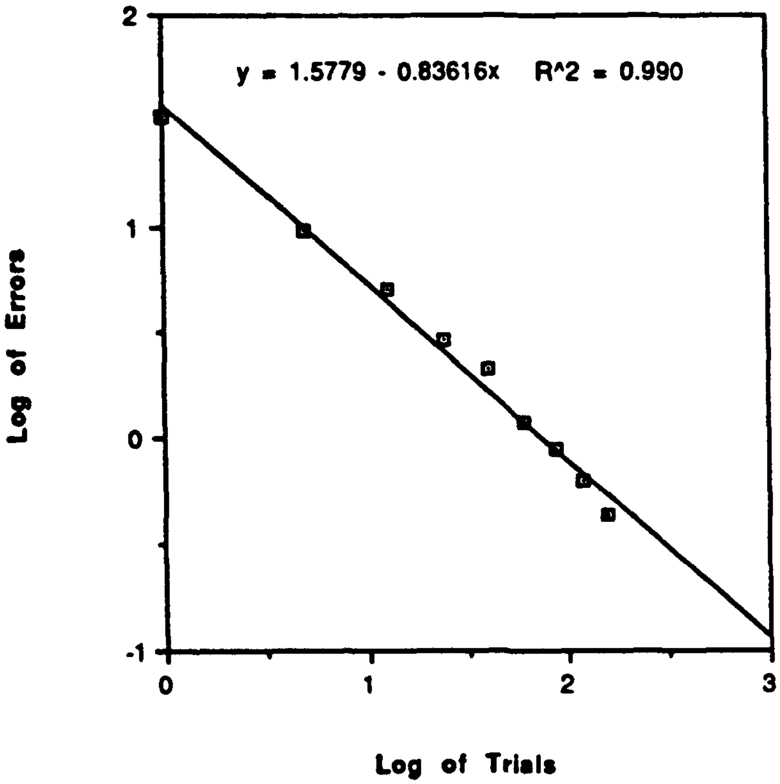

Figure 10. Performance as a function of trials.

realistic is this assumption? The assumption is true in *approximately uniform* task environments. By approximately uniform I mean that the average branching factor in a small neighborhood around a search state is equal for all states in the search space. If this is true and if performance is plotted as a function of learning events instead of as a function of trials, then the results should be a linear relationship with negative slope. Figure 11 shows the results from a simulation run in which HS was given repeated practiced on a particular Lewis problem. When performance is plotted as a function of learning events, the result approximates a negative linear relationship, indicating that the chemistry environment is, in fact, approximately uniform. An empirical test of the prediction that human learning is linear in the number of learning events (in this task environment) is possible in principle but requires a method for identifying learning events in human data.

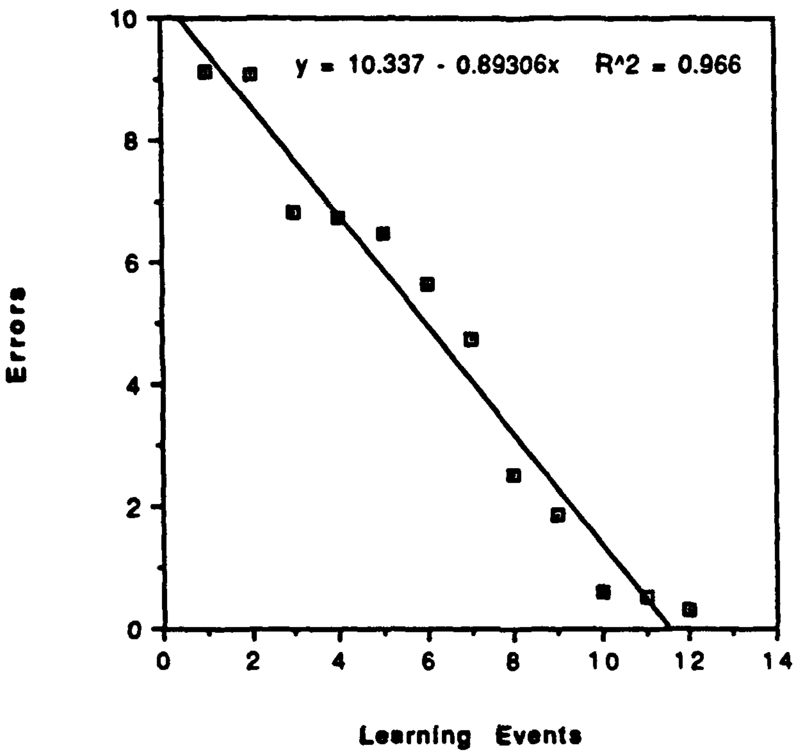

Figure 11. Performance as a function of learning events.

Transfer of Training

Background. Knowledge must be applicable in other situations than the one in which it was learned in order to be useful, but many laboratory studies have recorded little or no transfer of procedural knowledge even between isomorphic problems (Cormier & Hagman, 1987; Singley & Anderson, 1989, Chap. 1). Although many models of learning try to elucidate the mechanism of transfer (Ohlsson, 1987a; Singley & Anderson, 1989), the empirical data imply that *the main task for a transfer model is to elucidate why transfer of training does not occur.* In spite of the negative findings, psychologists keep trying to identify conditions that produce transfer, presumably because the findings strongly contradict our experience of ourselves as creatures with general and flexible competence. A second task for a theory of skill acquisition is to resolve this appearent contradiction between the laboratory findings and our intuitive self-understanding.

The production system hypothesis solves the first of these explanatory tasks. If procedural knowledge is encoded in production rules and if the rules required to solve a training task A are different from the rules required to solve a target task B, then practice on A will not affect the amount of learning required to master B, which is the typical laboratory result. Production rules are task specific, so they do not transfer.

From the point of view of common sense, the lack of transfer of training between isomorphic problems is particularly puzzling. A series of experiments with different versions of Duncker's ray problem has shown that unless subjects are explicitly reminded of the training task, transfer to an isomorphic target task is limited (Gick & Holyoak, 1987, pp. 34-37). Other experiments have verified that people behave differently on isomorphs of the Tower of Hanoi problem (Hayes & Simon, 1977) as well as on different isomorphs of the so-called selection task (see Evans, 1982, Chap. 9, for a review).

According to the production system hypothesis, these results are to be expected. Production rules for moving disks between pegs cannot also transfer globes among monsters; production rules that split up and recombine X-rays cannot also split up and recombine army platoons; production rules that decide whether envelopes have the proper postage cannot also test abstract rules; and so on. Production rules contain variables, but they quantify over arguments to predicates, not over predicates. There is no reason to expect a production rule to facilitate the construction of other rules isomorphic to itself, particularly not if the intended isomorphism is unknown to the learner.

The non-transferability of procedural knowledge leaves us with a picture of human beings as brittle systems which can only solve the very tasks that they have practiced. If this is true, then how do we survive even a single day of normal life?

The first answer is that the zero transfer prediction must be moderated by the distinction between *far transfer*, in which the target task differs completely from the training task, and *near transfer*, in which the two tasks partially overlap. In far transfer situations (which include almost all instructionally relevant situations) there is no overlap in the production rules for the two tasks and the production system hypothesis predicts zero transfer. In the near transfer case, on the other hand, there are rules in the procedure for the training task which are identical to rules in the procedure for the target task. In this case, there will be a transfer effect. Singely and Anderson (1989) claim that the number of production rules shared between two tasks is a good predictor of the amount of transfer in near transfer situations.

The second and more important answer suggested by the present theory is that generality resides in a person's declarative knowledge rather than in his or her procedural knowledge. It is our concepts and beliefs about the world that transfer from one situation to another, rather than our skills. We understand how the world works well enough so that we are able to construct the procedural knowledge required by novel circumstances and conditions. We cope by generating new procedures, not by transferring old procedures to new situations.

This explanation suggests that psychologists have been studying the wrong paradigm. Transfer studies have focussed on pairs of tasks which have similar solutions. In the typical transfer experiment, the experimenter varies the degree of similarity between the *solution* to a training task and the *solution* to a target task and expects the amount of transfer to vary accordingly. The negative findings from studies of isomorphic problems command attention because the solutions to those problems are structurally identical and so ought to yield perfect transfer.

However, the hypothesis that generality resides in declarative knowledge implies that structural similarity between solution paths is irrelevant. The important factor is whether two skills share a common conceptual rationale. If the skills required to solve two tasks A and B can both be derived from a set of beliefs or abstract principles T, then knowing T should give the ability to solve both A and B. The fact that two different procedures have the same theoretical rationale does not imply that there is any formal or structural similarity between the problem solutions generated by those procedures. For example, a chemical analysis of an unknown compound and a synthesis of a particular substance are procedurally different, but both are based on the same theory of the composition of matter.

Simulating Transfer of Training. The skill acquisition literature contains few studies of procedurally different skills which have the same declarative rationale, but developmental psychologists have found a naturally-occurring instance of this type of situation. Gelman and Gallistel (1978) have argued that children learn to count sets of objects by deriving the correct counting procedure from their intuitive understanding of its rationale. They formulated the declarative knowledge required for correct counting into a set of well-defined counting principles and presented empirical evidence that children know these principles at the time they learn how to count. Knowledge of the counting principles should give the ability to construct not only the procedure for the standard counting task, but to solve two non-standard counting tasks as well: to count objects in a particular order, so-called *ordered counting*, and to count objects in such a way that a designated object is assigned a designated number (e., g., "count the objects so that the red object becomes the fifth one"), so-called *constrained counting*. The empirical evidence confirms that children can quickly generate the correct procedures for these non-standard counting tasks (Gelman & Meck, 1983, 1986; Gelman, Meck, & Merkin, 1986).

To simulate this situation, the HS model was given a problem space for the task of counting a given set of objects. The representation included symbols for objects, numbers, for relations between objects and numbers, and so on. The actions included to select an object, to select a number, and to assign a number to an object. Figure 12 summarizes the problem space for counting.

The model was given rules which knew how to apply the operators, but which did not know how to apply them correctly. Finally, the model was given the counting principles in the form of constraints. Figure 13 summarizes the counting simulation. More detailed reports are available in Ohlsson and Rees (1991a, 1991b).

The model was trained on each of the three procedures for standard counting, ordered counting, and constrained counting. The diagonal of Table 1 shows the results as reported in Ohlsson and Rees (1991b). The effort required was measured in two ways, by the number of production system cycles and by the number of learning events, i. e., rule revisions. The model learned each procedure in approximately the same

Representation
Symbols for objects, numbers, sets of objects, and associations between numbers and either objects or sets. Both objects and numbers can have the properties of being *first*, *current*, and *point of origin*, and numbers can have the property of being the *answer*. The relations represented are *correspondancy*, *set membership*, *successor*, and *temporal contiguity*.

Initial state
A set of objects to be counted.

Operators
Associate a number with an object, associate a number with a set, select a first object, select the next object, select the first number, select the next number, and shift focus.

Goal state
A number designated as the cardinality of the given set.

Figure 12. A problem space for counting.

number of learning events. This result illustrates the generality of declarative knowledge. A single set of abstract principles gave the model the ability to construct three different procedures, each procedure being, in a sense, derived from those principles during practice. The declarative knowledge transferred from one counting task to another, even though the tasks are procedurally different.

Switching between counting tasks is an instance of near transfer. Not all rules need to be revised. Hence, the theory predicts that there will be procedural transfer as well. To illustrate this, six transfer experiments were run with the model. In each experiment, the model first practiced one of the three counting tasks until it reached mastery and then it was switched to either of the other two tasks. The results are

Prior procedural knowledge
The system began with one rule for each operator. That rule applied the operator whenever possible, i. e., in every situation in which its applicability conditions were satisfied. The result was counting-like but chaotic behavior.

Prior declarative knowledge
The were 18 constraints that encode the counting principles as identified by Gelman and Gallistel (1978): The one-to-one mapping principle, the cardinal principle, and the stable order principle.

Training
The model was given unsupervised practice on sets of 3-5 objects.

Learning outcomes
The model learned a correct, general procedure for counting any set of objects, regardless of the size of the set and the type of objects involved. It also learned correct procedures for two non-standard counting tasks, *ordered counting* and *constrained counting*. Finally, the model transfered each of the three learned counting procedures to each of the other two counting tasks.

Figure 13. Summary of the counting simulation.

shown in the off-diagonal cells of Table 1. The model solved each transfer task successfully. The amount of transfer varied depending on the exact relations between the rules for the practice task and the rules for the target task. The model also predicts asymmetric transfer between some tasks. For example, the transfer from ordered to constrained counting was 0%, while the transfer from constrained to ordered counting was 75%. These predictions are, in principle, empirically testable, although the necessary data are not available at this time.

Table 1. The computational effort required by the HS model to learn each of three counting tasks (diagonal cells) and to solve each of six different transfer tasks (off-diagonal cells).[a]

Training task	Transfer task: Standard counting	Ordered counting	Constrained counting
Standard counting			
Rule revisions	12	2	2
Prod. sys. cycles	854	110	127
Ordered counting			
Rule revisons	1	11	11
Prod. sys. cycles	184	262	297
Constrained counting			
Rule revisions	0	3	12
Prod. sys. cycles	162	154	451

[a]Data taken from Ohlsson and Rees (1991b, Tables 1 and 2).

Contrary to Singley and Anderson (1989), the present theory does not imply that the amount of transfer is predictable from the number of overlapping production rules. Instead, the variable of interest is the amount of cognitive work that has to be performed in order to adapt the rules to the target task. A single rule from the training task might cause more than one type of error in the target task and need to be revised more than once, so the number of rules that need to be revised is probably too course a predictor variable. The present analysis suggests that the number of rule revisions is a better predictor.

Unlike the number of overlapping production rules, the number of rule revisions required to master the target task cannot be calculated from a static comparison of the two procedures. It is a function of the particular learning mechanism which carries out the revisions. To verify this, the ***knowledge compilation*** mechanism-- a cornerstone of the ACT model described by Anderson (1983)--was implemented within the HS architecture. According to the knowledge compilation hypothesis, declarative knowledge resides in long-term memory in a format similar to inference rules. Familiar problems are solved with production rules, but unfamiliar problems are solved by interpreting (in the computer science sense) the declarative knowledge. During interpretation, new production rules are constructed which eliminates the need to re-interpret the declarative knowledge on subsequent trials. Once the rules are constructed, they are composed into larger rules which solve the relevant task more efficiently. Unlike HS, the ACT model learns from successes, not from errors.

We did not implement the ACT architecture as described in Anderson (1983). Instead, we implemented the knowledge compilation mechanism within the HS architecture. The result was a version of HS which learns through knowledge compilation instead of through constraint violations. All other aspects of the HS architecture were kept the same. I shall refer to the HS architecture as the KC model when it learns through knowledge compilation. The upshot is that we have two simulation models, HS and KC, which learn in different ways but which are otherwise identical. This provides an opportunity to compare the behavioral predictions of the two learning mechanisms.

KC was given the counting principles in the form of declarative knowledge and was then run through the same set of learning experiments and transfer experiments as HS. Because the effort measures differed by an order of magnitude (KC was on the average ten times slower than HS), they have been converted into transfer scores. There are many different ways to measure transfer (Singely & Anderson, 1989, pp. 37-41). The index used here was

$$T = \frac{E_B - E_{B/A}}{E_B} * 100 \qquad (2)$$

where E_B is the cognitive effort required to master the target task B from scratch and $E_{B/A}$ is the effort required to master B given previous mastery of training task A. The T index can be interpreted as the proportion of the effort required to learn task B that is saved by first learning task A. It is equal to zero when practice on the training task A is of no help and it is equal to 100 when practice on the training task provides mastery of B with no further training. The index is negative if practice on task A increases the amount of effort required to master B.

The effort measures for the transfer experiments with the HS and KC models were converted to transfer scores. The results are shown in Table 2. The amount of transfer predicted by the HS model varied between 8 and 100 across tasks, while the transfer predicted by the KC

Table 2. Transfer scores for the HS and KC[a] models for each of six transfer tasks in the counting domain, based on both the number of rule revisions and the number of production system cycles.

Transfer from-to	Effort measure: Rule revisions HS	Rule revisions KC	Prod. sys. cycles HS	Prod. sys. cycles KC
Standard-Ordered	82	100	58	100
Standard-Constrained	83	72	72	63
Ordered-Standard	92	34	78	19
Ordered-Constrained	8	34	34	29
Constrained-Standard	100	97	81	81
Constrained-Ordered	67	97	41	73

[a]KC is an acronym for *knowledge compilation*.

model varied between 19 and 100. More importantly for present purposes, the two models made different transfer predictions for one and the same task. HS predicts a score of 92 for the transfer from ordered to standard counting, while the corresponding KC prediction is 34. More important still, the differences between tasks do not always go in the same direction for the two models. HS predicts that transfer from standard to ordered counting is easier than vice versa, while KC predicts the opposite. The results in Table 2 verify the fact that the amount of near transfer between two tasks cannot be predicted from a static analysis of the procedures for those tasks, but depends upon assumptions about learning.

APPLICATION TO INSTRUCTION

A good theory should have implications for practice. The natural application domain for a learning theory is the design of instruction. The instructional implications of the present theory include an explanation of why it is possible to learn from instruction, a rationale for the most common tutoring scenario, a prescription for effective tutoring messages, and a technology for evaluating instructional designs through simulated one-on-one tutoring.

Why Instruction is Possible

Why are people able to learn from instruction? Although the origin of cognitive capacities such as language and learning is almost completely unknown, it is likely that learning evolved before language. There are no mammalian species, and probably no lower organisms, which cannot learn, so the capacity to learn was almost certainly present in the hominids when they separated from the rest of the primates 4-10 millions of years ago.

Language, on the other hand, evolved later, perhaps very much later. McCrone (1992) summarizes the fossil evidence in the following way: "The high arch in the roof of the mouth that helps with voice production is about the only telltale sign of speech that shows up on a fossil skeleton. This arch did not start to appear until *Homo erectus* arrived about 1.5 million years ago, and even then the arch was slight. Judging from fossils, modern speech came along about 100,000 years ago when the earliest examples of *Homo sapiens* were starting to walk the earth." (p. 160-161)[3] One hundred thousand years is a short time in evolutionary terms. If this estimate is correct, then special-purpose brain mechanisms for learning from verbal instruction have had little time to evolve.

These two speculative but plausible hypotheses--that learning preceeded language and that language is too recent for special-purpose brain mechanisms for instruction to have evolved--imply that our ability to learn without instruction is primary and our ability to learn from instruction secondary and parasitic upon the former. A theory of learning from instruction should therefore explain how instruction feeds into learning mechanisms that evolved for the purpose of uninstructed learning.

The theory proposed in this chapter suggests such an explanation. The two functions of detecting and correcting errors can be computed by noticing contradictions and by inferring the conditions that produced them as described previously, but they can also be computed in other ways. Instruction works, the theory suggests, because being told that one has committed an error is functionally equivalent to detecting the error oneself and because being told the cause of an error is functionally equivalent to figuring out the cause oneself. Learning from instruction is possible because instructional messages enter into the learning process in the same way, functionally speaking, as declarative knowledge retrieved from long-term memory.

A Rationale for One-on-One Tutoring

The theory proposed in this chapter implies that there are three major felicity conditions (VanLehn, 1990, p. 23) for effective instruction in cognitive skills: (a) instruction should be offered during ongoing practice, (b) instruction should alert the learner to errors, and (c) instruction should identify the conditions which caused the error. The type of in-

[3] See Lyons (1988, p. 153) for a different interpretation of the evidence.

struction that satisfies these three conditions is entirely familiar. In one-on-one tutoring, the teacher watches as the learner practices, points out errors, and helps the learner correct them. The present theory selects as most felicitous precisely that type of instruction which the empirical data show is most effective (Bloom, 1984).

Intelligent tutoring systems are typically designed to teach cognitive skills (Psotka, Massay, & Mutter, 1988; Sleeman & Brown, 1982). Perhaps the most successful line of intelligent tutoring systems are the so-called model tracing tutors developed by John Anderson and co-workers (Anderson et al., 1987, 1990). Skill training tutors in general and model tracing tutors in particular conform closely to the three felicity conditions: They give feedback in the context of practice, they alert the learner to errors, and they help the learner to correct the error.

The design of the model tracing tutors is said to be derived from the ACT theory of learning (Anderson et al., 1987). However, none of the six learning mechanisms described in various versions of the ACT theory--analogical transfer, discrimination, generalization, proceduralization, rule composition, and strengthening--can take a tutoring message as input and revise a faulty production rule accordingly. Analogical transfer generates task relevant activity by relating the current problem to an already solved problem; rule composition creates more efficient rules by combining existing (hopefully correct) rules; strengthening increases the probability that a (hopefully correct) rule will be retrieved. These three learning mechanisms can neither take a tutoring message as input nor revise an existing rule. Generalization and discrimination (which do not loom large in expositions of the ACT theory) revise existing rules, but cannot take a tutoring message as input. Proceduralization generates new rules on the basis of verbal input, but cannot correct existing rules. Taken literally, ***the ACT theory predicts that it is impossible to learn from the teaching scenario embodied in the model tracing tutors***. Unless people can learn in other ways than those described in the ACT theory, they have no cognitive mechanisms for learning from feedback messages about errors.

To highlight the contrast between the implications of the ACT theory and the design of the model tracing tutors, consider what an intelligent tutoring system derived from the ACT theory might be like. In order to facilitate analogical transfer, such a system might keep a record of the problems the student has solved in the past and suggest possible analogies when the student hesitates. Such a system might repeat the task instructions from time to time to give the student a chance to re-proceduralize them. It might sequence practice problems in such a way that rule composition, generalization, and discrimination are facilitated. Finally, it might provide opportunities to exercise already acquired components of the target skill in order to increase their strengths. However, a tutoring system derived from the ACT theory would have no reason to alert the learner to errors and give help in correcting them.

The model tracing tutors and most other skill training systems conform to the design that follows from the theory presented in this chapter: They help the learner detect and correct errors. The instructional success of such tutors provide support for the hypothesis that error correction is the natural ***modus operandi*** of skill acquisition. If it were not, those tutors would not be effective but empirical evaluations show that they are (Anderson et at., 1990, pp. 30-33). In short, the present theory provides a rationale for the teaching scenario adopted by designers of intelligent tutoring systems and in turn receives empirical support from the instructional success of such systems.

Deriving the Content of Instruction from Theory

The content of feedback messages is the Achilleus heel of skill-monitoring tutoring systems. Delivering feedback messages is the major instructional action of such a system, so its instructional effectiveness depends crucially on the content of those messages. Until now there has been no theory for how to formulate feedback messages. Such messages are typically pre-formulated texts and they are written in the same way as other instructional materials: The instructional designer makes a guess about what might work based on his or her understanding of the subject matter.[4] In spite of the strong claims about the tight relation between the ACT theory and the model-tracing tutors (Anderson et al., 1987), this is true of those tutors as well. No existing tutoring system

[4]See Moore and Ohlsson (1992) and Reiser et al. (1991) for exceptions.

derives the content of its tutoring messages from assumptions about learning.

The learning theory proposed here implies that tutoring messages should help the student identify those properties of the current problem state which indicate that an error has been committed, so that he or she can detect his or her errors without help in the future. The general form for this type of tutoring message is "you can tell that you just made an error, because of P", where P is some conjunction of easily accessed properties of the problem state produced by the erroneous action. Unless the learner can detect his or her errors, he or she cannot learn from them.

More importantly, tutoring messages should help the learner correct his or her errors. To do so, a message must identify those properties of the immediately preceeding problem state that constitute counterindications to the problem solving step that the student took. A problem solving step A is typically correct in some situations but wrong in others. The task of the student is to figure out when, i. e., in which situations, doing A is right and when it is wrong. If doing A in situation S is incorrect, then the corresponding tutoring message should have the general form "when such-and-such conditions are the case, A is not the right thing to do". The conditions mentioned in the message should refer to the immediately preceeding problem situation, not to the situation in which the error was discovered. The student needs to learn to avoid the error, i. e., to act differently in the situation in which he or she decided to do A. The tutoring system should therefore back up and explain what makes A the wrong choice in that situation.

These prescriptions rule out some types of feedback messages which are commonly used in tutoring systems. For example, it is intuitively plausible that if a student takes step A when he or she should have taken step B, then it helps to print a message of the form "you did A but you should have done B." According to the present theory, however, this type of feedback message is likely to be ineffective, because it does not specify the conditions under which either A or B should or should not be done. Instruction should focus on the conditions of actions, not on the actions themselves. A second common type of feedback message explains what is wrong with the situation in which the error was discovered, i. e., why the error is an error. This might increase the student's understanding of the domain but it is unlikely to help him or her acquire the target skill, because it does not tell him or her how to avoid the error. A feedback message should focus on the situation in which the student decided to do A, not on the situation produced by doing A.

Simulating One-on-One Tutoring

The HS model can be interpreted as a model of learning from tutoring, with the constraints playing the role of tutoring messages. It is a matter of interpretation whether the constraints correspond to knowledge items retrieved from memory, conclusions from inference chains, or tutoring messages received through the language comprehension channel. According to the theory proposed here, these three types of knowledge elements enter into the learning process in the same way.

A runnable simulation of learning from tutoring opens up novel possibilities. We can evaluate alternative instructional designs by teaching them to the model and measuring the amount of computational work it has to expand to learn the target skill under different circumstances. If the model can reach mastery with less work under one instructional design or tutoring regime than another, then that is evidence that the former is the better design.

To explore this possibility the HS model was tutored in subtraction. The simulation experiment followed the common classroom tactic of teaching the procedure for canonical subtraction problems, i. e., problems in which each subtrahend digit is smaller than the minuend digit in the same column, and to introduce the procedure for how to handle non-canonical columns, i. e., columns in which the subtrahend digit is larger than the minuend digit, once the procedure for canonical subtraction has been mastered (Leinhardt, 1987; Leinhardt & Ohlsson, 1990). The simulation experiment followed this pedagogical tactic in that the model was first given a procedure for non-canonical subtraction and was then tutored in canonicalization.

Two different HS models of canonical subtraction were implemented. One model, called the ***high-knowledge model***, was built around a representation of the place value meaning of digits. In this representation the digit 3 was represented as (3 * 10) if it appeared in the second column to the right, as (3 * 100) if it appeared in the third column, and so on. The operations by which this representation was manipulated correspond to mathematically motivated operations on numbers. The high-knowledge model was intended to simulate skill acquisition in the context of conceptual understanding of place value.

The second model, called the *low knowledge model*, was built around a representation in which a subtraction problem is a two-dimensional array of digits. In this representation, the digit 3 was represented as the digit 3 regardless of its position in the problem display. The operations by which this representation was manipulated correspond to physical operations on digits rather than conceptual or mathematical operations on numbers. The low knowledge model was intended to simulate rote learning of subtraction. Figure 14 summarizes the problem space for subtraction. The reader is referred to Ohlsson, Ernst, and Rees (1992) for a full account.

Both the high and the low knowledge models were tutored in the regrouping algorithm taught in American schools. In this method non-canonical columns are handled by incrementing the minuend of the non-canonical column and performing a corresponding decrement on the minuend in the next column to the left. Both models were also tutored in the equal addition algorithm taught in some European schools. In this method non-canonical columns are handled by incrementing the minuend in the non-canonical column and decrementing the subtrahend in the next column to the left. The simulation experiment thus followed a 2-by-2 design, with two levels of knowledge paired with two different target skills.

The procedure for tutoring the model were similar to those involved in tutoring a human student. The programmer in charge of the system watched while the model tried to solve a non-canonical problem, spotted errors, halted the model, and typed in a constraint (tutoring message) intended to correct the observed error. When the model had attained mastery, it was reinitialized and run again with all the constraints in place simultaneously, to verify that they were indeed sufficient to produce correct performance. This tutoring scenario was carried out four

Representations
Two different representations for subtraction were created. (a) The procedural or ***low knowledge*** representation contained symbols for written digits, perceived digits, spatial locations, scratch marks, decrements, and increments. (b) The conceptual or ***high knowledge*** representation contained, in addition, symbols for subtraction problems, numbers, place values, links between numbers and digits, relations between numbers, and answers.

Initial state
A subtraction problem.

Operators
Look at a digit, move the eye to another digit, write a digit, cross out a digit, assert the answer, recall number fact, create a working memory schema, and revise a working memory schema.

Goal state
A number designated as the answer to the subtraction problem.

Figure 14. A problem space for subtraction.

times, once for each combination of knowledge level and target skill. The amount of computational work required to attain mastery in each condition was recorded. Figure 15 summarizes the subtraction simulation.

Table 3 shows the number of production system cycles and the number of rule revisions required for HS to attain mastery in each of the four conditions. There are two main results. The high knowledge model required more work to attain mastery than the low knowledge model. This is true for both canonicalization procedures and for both effort

Prior procedural knowledge
The system knew at the outset how to solve a canonical subtraction problem, i. e., a problem in which the subtrahend digit is smaller than the minuend digit in every column.

Training
The model was tutored in how to handle non-canonical problems. It executed its procedure for canonical problem until it made an error. It was then halted and given a constraint that was intended to allow it to correct the error.

Learning outcome
The model learned two different procedures for non-canonical columns, namely regrouping and equal addition, with both the low knowledge and the high knowledge representations.

Figure 15. Summary of the subtraction simulation.

measures. The reason for this result is that the high knowledge model had a more elaborate representation. It requires more cognitive operations to create and update a more elaborate representation and each operation must be guided by some production rule. Hence, the high knowledge model had more to learn.

The second result is that learning the regrouping procedure required more work than learning the equal addition procedure in both the high knowledge and the low knowledge conditions. The reason for this result is that the control of the regrouping procedure becomes complicated when it is necessary to regroup the minuend recursively to handle blocking zeroes, i. e., minuend zeroes immediately to the left of a non-canonical column. The augmenting procedure is not affected by the number of blocking zeroes. The regrouping procedure also requires more complicated visual attention allocation.

Table 3. The computational effort required for the HS model to master regrouping and augmenting with either a high knowledge or a low knowledge representation.[a]

Method learned	High knowledge: Cycles	High knowledge: Revisions	Low knowledge: Cycles	Low knowledge: Revisions
Regrouping				
W/o blocking zeroes[b]	940	23	449	16
With blocking zeroes	1815	32	794	24
Augmenting				
W/o blocking zeroes	862	20	687	18
With blocking zeroes	862	20	687	18

(Type of representation spans High knowledge and Low knowledge columns.)

[a]Data taken from Ohlsson, Ernst, and Rees (1992, Table 2).
[b]I. e., minuend zeroes to the left of a non-canonical column.

Both of these results were unexpected because they contradict the common belief among mathematics educators that regrouping is easier to learn, particularly in the high knowledge condition. This belief is based on empirical investigations carried out in the pre-Word War II era (Brownell, 1947; Brownell & Moser, 1949). A detailed discussion of these simulation results and their relation to the empirical research has been presented elsewhere (Ohlsson, 1992a).

This simulation exercise demonstrates that runnable models of learning from instruction creates new relations between learning theory and instructional design (Ohlsson, 1992a). Instead of deriving general design principles from the learning theory, as suggested by Bruner (1966) and later by Glaser (1976, 1982), we can evaluate an instructional design directly by teaching it to a simulation model. This technology has the potential to allow instructional designers to do formative evaluation without leaving their desks (Ohlsson, 1992b).

This simulation exercise also demonstrates that the application of learning theory to education requires a formal analysis of instruction. *A model of learning cannot have implications for instruction unless it contains learning mechanisms which take instruction, suitably formalized, as one of their inputs*. Computational analysis of instruction has barely begun. Some early Artificial Intelligence systems explored how a system can learn from advice and instructions (Hayes-Roth, Klahr, & Mostow, 1981; Mostow, 1983; Rychener, 1983), but the problem appears to have disappeared from the research agenda of the machine learning community. The proceduralization mechanism in the ACT model (Anderson, 1983) was a first attempt to formalize this problem in a psychological context. Although the proceduralization mechanism explains how the learner constructs new rules on the basis of task instructions, it does not explain how the learner revises existing rules on the basis of feedback messages. The Sierra and Cascade models described by VanLehn (1990) and VanLehn and Jones (this volume) learn from solved examples--a common form of instruction--but they cannot take tutoring messages as inputs. The HS model constitutes a modest first step towards a formal theory of how tutoring messages received during skill practice are translated into mental code for the target skill.

SUMMARY AND CONCLUSIONS

The theory proposed in this chapter is formulated in terms of cognitive functions instead of information processing mechanisms. The function of learning to solve an unfamiliar task is analyzed into two subfunctions: To generate learning opportunities and to construct new procedural knowledge. The latter function is in turn broken down into two subfunctions: To detect incorrect problem solving steps and to correct the procedural knowledge that generated them. To detect errors requires a comparison between the current problem state with prior knowledge. To correct an error, finally, involves identifying the conditions under which the error appears and constraining the faulty decision rule accordingly. The main claim of the theory is that this is the right functional breakdown of skill acquisition.

How does this theory explain the role of prior knowledge in skill acquisition? Domain knowledge is not needed to generate task relevant actions. Weak methods can generate behavior even in the absence of any knowledge about the task. The functions for which knowledge is needed are to detect and correct errors. Incorrect or incomplete task procedures are likely to produce situations which contradict what ought to be true in the particular domain. Domain knowledge increases the probability that the learner recognizes that he or or she has made an error. To correct the error presupposes the ability to identify the conditions under which that error will appear. This might require complicated reasoning about the domain. Prior knowledge increases the probability that the learner identifies the causes of errors correctly.

Each of the functions postulated in the theory can be computed by many different information processing mechanisms. In the particular implementation of the theory described in this chapter, task relevant behavior is generated by forward search through the problem space. Errors are detected by matching constraints against problem states with a pattern matcher. The conditions that produced a particular error are identified by regressing the match between a constraint and a state through a production rule. Errors are corrected by adding the conditions that produce them to the left-hand sides of decision rules. Other implementations of the functional theory are possible.

The simulation model generates several quantitative predictions about two classical problems in learning theory. First, it predicts that skill acquisition is negatively accelerated. More precisely, it predicts that the learning curve follows a so-called power law. Second, the theory predicts zero transfer in far transfer situations. It also predicts that the amount of transfer in near transfer situations depends upon the particular tasks involved and that transfer might be asymmetrical, i. e., that there might be either more transfer from task B to task A than from task A to task B. Finally, the model predicts that the richer the representation of the task to be learned, the more cognitive effort is needed to attain mastery.

With respect to the practical problem of designing computer-based instruction in cognitive skills, the present theory provides a rationale and an explanation for the effectiveness of one-on-one tutoring, the main teaching scenario embodied in current intelligent tutoring systems. Tutoring (by computer or by human) works, the theory of claims, because tutoring messages provide an alternative way to become aware of errors and an alternative source of information about the conditions under which the errors occur.

According to the present theory, tutoring messages can help the learner in two ways. First, to help the learner detect his or her own errors, tutoring messages should point out those properties of a problem state which indicate that an error has occurred. Second, to help the learner correct his or her errors, tutoring messages should identify those properties of a problem state which indicate that an error will occur if such and such an action is executed.

The theory proposed here is obviously incomplete. People undoubtedly learn from their errors, but they also learn from their successes. The theory needs to be extended with assumptions about how people learn from correct problem solving steps. It is not clear which of those predictions will remain constant if the model is augmented with additional learning mechanisms. The interaction between multiple learning mechanisms is a high-priority issue for computational learning theories. In past work, I combined a method for learning from error (discrimination) with two methods for learning from success (generalization and subgoaling). The resulting model learned to solve simple puzzle tasks (Ohlsson, 1987a), but it threw no light on the problem of prior knowledge.

The problem of how prior knowledge impacts learning is central for the study of skill acquisition. The outcome of practice is always a function of both the learner's prior knowledge about the domain and the new information that becomes available during practice. Any viable learning theory must describe the cognitive mechanism that interfaces those two knowledge sources. The fate of the theory proposed here will ultimately be determined by comparative evaluations with alternative computational theories of the function of prior knowledge in learning, once such alternative theories become available.

REFERENCES

Abelson, R. P., Aronson, E., McGuire, W. J., Newcomb, T. M., Rosenberg, M. J., & Tannenbaum, P. H., (Eds.), (1968). *Theories of cognitive consistency: A sourcebook.* Chicago, IL: Rand McNally.

Anderson, J. R. (1976). *Language, memory, and thought.* Hillsdale, New Jersey: Erlbaum.

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J. R. (1986). Knowledge compilation: The general learning mechanism. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell, (Eds.), *Machine learning: An artificial intelligence approach* (Vol. 2, pp. 289-310). Los Altos, CA: Kaufmann.

Anderson, J. R. (1987). Skill acquisition: Compilation of weak-method problem solutions. *Psychological Review, 94*, 192-210.

Anderson, J. R. (1990). *The adaptive character of thought.* Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Erlbaum. In press.

Anderson, J. R., Boyle, C. F., Corbett, A. T., & Lewis, M. W. (1990). Cognitive modeling and intelligent tutoring. *Artificial Intelligence, 42*, 7-49.

Anderson, J. R., Boyle, C. F., Farrell, R., & Reiser, B. J. (1987). Cognitive principles in the design of computer tutors. In P. Morris, (Ed.), *Modeling cognition.* New York, NY: Wiley.

Anderson, J. R., Kline, P. J., & Beasley, C. M., Jr. (1979). A general learning theory and its application to schema abstraction. In G. H. Bower, (Ed.), *The psychology of learning and motivation: Advances in research and theory* (Vol. 13, pp. 277-318). New York, NY: Academic Press.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2, 396-408.

Angluin, D., & Smith, C. H. (1983). Inductive inference: Theory and methods. *Computing Surveys, 15*, 237-269.

Anzai, Y., & Simon, H. A. (1979) The theory of learning by doing. *Psychological Review, 86*, 124-140.

Berkson, W., & Wettersten, J. (1984). *Learning from error: Karl Popper's psychology of learning*. La Salle, IL: Open Court.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher, 13(6)*, 4-16.

Brownell, W. A. (1947). An experiment on "borrowing" in third-grade arithmetic. *Journal of Educational Research, 41*, 161-263.

Brownell, W. A., & Moser, H. E. (1949). *Meaningful vs. mechanical learning: A study in Grade III subtraction*. Durham, NC: Duke University Press.

Bruner, J. S. (1966). *Toward a theory of instruction*. Cambridge, MA: Harvard University Press.

Confrey, J. (1990). A review of the research on student conceptions in mathematics, science, and programming. In C. B. Cazden, (Ed.), *Review of Research in Education* (Vol. 1, pp. 3-56). Washington, D. C.: American Educational Research Association.

Cormier, S., & Hagman, J., (Eds.), (1987). *Transfer of learning: Contemporary research and applications*. New York, NY: Academic Press.

Duhem, P. (1991/1914). *The aim and structure of physical theory* (2nd ed.). Princeton, NJ: Princeton University Press.

Ebbinghaus, H. (1964/1885). *Memory: A contribution to experimental psychology*. New York, NY: Dover.

Evans, J. St. B. T. (1982). *The psychology of deductive reasoning*. London, UK: Routledge & Kegan Paul.

Feigenbaum, E. A., & Feldman, J. (1963). *Computers and thought*. New York, NY: McGraw-Hill.

Festinger, L. (1957). *A theory of cognitive dissonance*. Standford, CA: Stanford University Press.

Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence, 2*, 189-208.

Forgy, C. L. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence, 19*, 17-37.

Gardenfors, P. (1988). *Knowledge in flux: Modeling the dynamics of epistemic states*. Cambridge, MA: MIT Press.

Gelman, R., & Gallistel, C. (1978). *The child's understanding of number*. Cambridge, MA: Harvard University Press.

Gelman, R., & Meck, E. (1983). Preschoolers' counting: Principle before skill. *Cognition, 13*, 343-359.

Gelman, R., & Meck, E. (1986). The notion of principle: The case of counting. In J. H. Hiebert, (Ed.), *Conceptual and procedural knowledge: The case of mathematics* (pp. 29-57). Hillsdale, NJ: Erlbaum.

Gelman, R., Meck, E., & Merkin, S. (1986). Young children's numerical competence. *Cognitive Development, 1*, 1-29.

Gensler, W. J. (1987). Impossibilities in chemistry: Their rise, nature, and some great falls. In P. J. Davis and D. Park, (Eds.), *No way: The nature of the impossible* (pp. 73-89). New York, NY: Freeman.

Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology, 15*, 1-38.

Gick, M. L., & Holyoak, K. J. (1987). The cognitive basis of knowledge transfer. In S. M. Cormier & J. D. Hagman, (Eds.), *Transfer of learning: Contemporary research and applications* (pp. 183-215). San Diego, CA: Academic Press.

Glaser, R. (1976). Components of a theory of instruction: Toward a science of design. *Review of Educational Research, 46*, 1-24.

Glaser, R. (1982). Instructional psychology: Past, present, and future. *American Psychologist, 37*, 292-305.

Hall, R. (1988). Learning by failing to explain: Using partial explanations to learn in incomplete or intractable domains. *Machine Learning, 3*, 45-77.

Hayes, J. R., & Simon, H. A. (1977). Psychological differences between problem isomorphs. In J. Castellan, D. B. Psioni, and G. Potts, (Eds.), *Cognitive theory* (Vol. 2, pp. 21-41). Hillsdale, NJ: Erlbaum.

Hayes-Roth, F., Klahr, P., & Mostow, D. (1981). Advice taking and knowledge refinement: An iterative view of skill acquisition. In J.

Anderson, (Ed.), *Cognitive skills and their acquisition* (pp. 231-253). Hillsdale, NJ: Erlbaum.

Hewson, P. W., & Hewson, M. G. A. (1984). The role of conceptual conflict in conceptual change and the design of science instruction. *Instructional Science, 13*, 1-13.

Holland, J., Holyoak, K., Nisbett, R., & Thagard, P. (1986). *Induction: The processes of inference, learning, and discovery*. Cambridge, MA: MIT Press.

Hull, C. L. (1943). *Principles of behavior*. New York, NY: Appleton-Century-Crofts.

Judd, C. H. (1908). The relation of special training and general intelligence. *Educational Review, 36*, 28-42.

Katona, G. (1940). *Organizing and memorizing: Studies in the psychology of learning and teaching*. Morningside Heights, NY: Columbia University Press.

Kieras, D. E., & Bovair, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science, 8*, 255-273.

Klahr, D., Langley, P., & Neches, R., (Eds.), (1987). *Production system models of learning and development*. Cambridge, MA: MIT Press.

Kocabas, S. (1991). Conflict resolution as discovery in particle physics. *Machine Learning, 6*, 277-309

Kuhn, T. (1970). *The structure of scientific revolutions* (2nd ed.) Chicago, IL: University of Chicago Press.

Laird, J. (1986). Universal subgoaling. In J. Laird, P. Rosenbloom, & A. Newell, *Universal subgoaling and chunking: The automatic generation and learning of goal hierarchies* (pp. 3-131). Boston, MA: Kluwer.

Lane, N. (1987). *Skill acquisition rates and patterns: Issues and training implications*. New York, NY: Springer-Verlag.

Langley, P. (1985). Learning to search: From weak methods to domain-specific heuristics. *Cognitive Science, 9*, 217-260.

Langley, P. (1987). A general theory of discrimination learning. In Klahr, D., Langley, P., & Neches, R., (Eds.), (1987). *Production system models of learning and development* (pp. 99-161). Cambridge, MA: MIT Press.

Leinhardt, G. (1987). Development of an expert explanation: An analysis of a sequence of subtraction lessons. *Cognition and Instruction, 4*, 225-282.

Leinhardt, G., & Ohlsson, S (1990). Tutorials on the structure of tutoring from teachers. *Journal of Artificial Intelligence in Education,* 2, 21-46.

Lyons, J. (1988). Origins of language. In A. Fabian, (Ed.), *Origins: The Darwin College lectures* (pp. 141-166). Cambridge, MA: Cambridge University Press.

Mazur, J., & Hastie, R. (1978). Learning as accumulation: A reexamination of the learning curve. *Psychological Bulletin, 85*, 1256-1274.

McCrone, J. (1992). *The ape that spoke: Language and the evolution of the human mind.* New York, NY: Avon Books.

McDermott, D., & Doyle, J. (1980). Non-monotonic logic I. *Artificial Intelligence, 13*, 41-72.

Minton, S. (1988). *Learning search control knowledge: An explanation-based approach.* Boston, MA: Kluwer.

Moore, J. D., & Ohlsson, S. (1992). Educating migraine patients through on-line generation of medical explanations. *Proceedings of the 14h Annual Conference of the Cognitive Science Society.* July 29 - August 1, Bloomington, Indiana, USA. In press.

Mostow, D. J. (1983). Machine transformation of advice into a heuristic search procedure. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell, (Eds.), *Machine learning: An artificial intelligence approach* (pp. 367-404). Palo Alto, CA: Tioga.

Mostow, J., & Bhatnager, N. (1987). Failsafe--a floor planner that uses EBG to learn from its failures. In *Proceedings of the International Joint Conference on Artificial Intelligence* (pp. 249-255), Milan, Italy: IJCAI.

Mostow, J., & Bhatnager, N. (1990). *Heuristic detection, explanation, and avoidance of failures as an approach to intractability in explanation-based learning of search control knowledge* (Technical Report No. 171). Rutgers, NJ: Rutgers University.

Murray, F. B., Ames, G. J, & Botvin, G. J. (1977). Acquisition of conservation through cognitive dissonance. *Journal of Educational Psychology, 69*, 519-527.

Newell, A. (1969). Heuristic programming: Ill-structured problems. In J. Aronofsky, (Ed.), ***Progress in operations research*** (Vol. III, pp. 360-414). New York, NY: Wiley.

Newell, A. (1980). Reasoning, problem solving and decision processes: The problem space as a fundamental category. In R. Nickerson, (Ed.), ***Attention and performance VIII*** (pp. 693-718). Hillsdale, NJ: Erlbaum.

Newell, A. (1990). ***Unified theories of cognition***. Cambridge, MA: Havard University Press.

Newell, A., & Rosenbloom, P. (1981). Mechanisms of skill acquisition and the law of practice. In J. Anderson, (Ed.), ***Cognitive skills and their acquisition*** (pp. 1-55). Hillsdale, NJ: Erlbaum.

Newell, A., & Simon, H. (1972). ***Human problem solving***. Englewood Cliffs, NJ: Prentice-Hall.

Nilsson, N. J. (1980). ***Principles of artificial intelligence***. Palo Alto, CA: Tioga.

Ohlsson, S. (1987a). Transfer of training in procedural learning: A matter of conjectures and refutations? In L. Bolc, (Ed.), ***Computational models of learning*** (pp. 55-88). Berlin, West Germany: Springer-Verlag.

Ohlsson, S. (1987b). Truth versus appropriateness: Relating declarative to procedural knowledge. In D. Klahr, P. Langley, & R. Neches, (Eds.), ***Production system models of learning and development*** (pp. 287-327). Cambridge, MA: MIT Press.

Ohlsson, S. (1992a). Artificial instruction. A method for relating learning theory to instructional design. In P. Winne & M. Jones, (Eds.), ***Foundations and frontiers in instructional computing systems***. New York, NY: Springer-Verlag. In press.

Ohlsson, S. (1992b). The impact of cognitive theory on the practice of courseware authoring. In R. Lewis, (Ed.), ***Authoring environments for computer-based courseware.*** New York, NY: Springer-Verlag. In press.

Ohlsson, S. (1992c). The learning curve for writing books: Evidence from Professor Asimov. *Psychological Science*. In press.

Ohlsson, S., Ernst, A. M., & Rees, E. (1992). The cognitive complexity of doing and learning arithmetic. *Journal of Research in Mathematics Eduation*. In press.

Ohlsson, S., & Rees, E. (1991a). The function of conceptual understanding in the learning of arithmetic procedures. *Cognition and Instruction, 8*, 103-179.

Ohlsson, S., & Rees, E. (1991b). Adaptive search through constraint violation. *Journal of Experimental and Theoretical Artificial Intelligence, 3*, 33-42.

Pearl, J. (1984). *Heuristics: Intelligent search strategies for computer problem solving*. Reading. MA: Addison-Wesley.

Piaget, J. (1971). *Biology and knowledge: An essay on the relation beween organic regulations and cognitive processes*. Edingburgh, UK: Edinburgh University Press.

Piaget, J. (1985). *The equilibrium of cognitive structures: The central problem of intellectual development*. Chicago, IL: University of Chicago Press.

Popper, K. (1972/1935). *The logic of scientific discovery*. London, UK: Hutchinson.

Posner, G. J., Strike, K. A., Hewson, P. W., & Gertzog, W. A. (1982). Accomodation of a scientific conception: Toward a theory of conceptual change. *Science Education, 66*, 211-227.

Psotka, J., & Massey, L. D., & Mutter, S. A., (Eds.), (1988). *Intelligent tutoring systems: Lessons learned*. Hillsdale, NJ: Erlbaum.

Quine, W. V., & Ullian, J. S. (1978). *The web of belief* (2nd ed.). New York, NY: Random House.

Reason, J. (1990). *Human error*. Cambridge, UK: Cambridge University Press.

Reif, F. (1987). Interpretation of scientific or mathematical concepts: Cognitive issues and instructional implications. *Cognitive Science, 11*, 395-416.

Reiser, B. J., Kimberg, D. Y., Lovett, M. C., & Ranney, M. (1991). Knowledge representation and explanation in GIL, an intelligent tutor for programming. In J. H. Larkin & R. W. Chabay, (Eds.), *Computer assisted instruction and intelligent tutoring systems: Shared issues and complementary approaches* (pp.111-149). Hillsdale, NJ: Erlbaum.

Resnick, L. B., & Omanson, S. F. (1987). Learning to understand arithmetic. In R. Glaser, (Ed.), *Advances in instructional psychology* (Vol 3, pp. 41-95). Hillsdale, NJ: Erlbaum.

Rose, D., & Langley, P. (1986). Chemical discovery as belief revision. *Machine Learning, 1*, 423-452.

Rosenbloom, P. (1986). The chunking of goal hierarchies. A model of practice and stimulus-response compatibility. In J. Laird, P. Rosenbloom, & A. Newell, *Universal subgoaling and chunking: The automatic generation and learning of goal hierarchies* (pp. 135-1282). Boston, MA: Kluwer.

Rychener, M. D. (1983). The instructable production system: A retrospective analysis. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell, (Eds.), *Machine learning: An artificial intelligence approach* (pp. 429-460). Palo Alto, CA: Tioga.

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

Schank, R. (1986). *Explanation patterns: Understanding mechanically and creatively*. Hillsdale, NJ: Erlbaum.

Singely, M., & Anderson, J. (1989). *The transfer of cognitive skill*. Cambridge, MA: Harvard University Press.

Sleeman, D., & Brown, J. S., (Eds.), (1982). *Intelligent tutoring systems*. New York, NY: Academic Press.

Smith, E. E., & Goodman, L. (1984). Understanding written instructions: The role of an explanatory schema. *Cognition and Instruction, 1*, 359-396.

Solomons, T. (1988). *Organic chemistry* (4th ed.). New York, NY: Wiley.

Sorum, C. H., & Boikess, R. S. (1981). *How to solve general chemistry problems*. Englewood Cliffs, NJ: Prentice-Hall.

VanLehn, K. (1982). Bugs are not enough: Empirical students of bugs, impasses and repairs in procedural skills. *Journal of Mathematical Behavior, 3*, 3-71.

VanLehn, K. (1988). Toward a theory of impasse-driven learning. In H. Mandl & A. Lesgold, (Eds.), *Learning issues for intelligent tutoring systems* (pp. 19-41). New York, NY: Springer Verlag.

VanLehn, K. (1990). *Mind bugs: The origins of procedural misconceptions*. Cambridge, MA: MIT Press.

Winograd, T. (1975). Frame representations and the declarative/procedural controversy. In D. Bobrow and A. Collins, (Eds.), *Representation and understanding: Studies in cognitive science* (pp. 185-210). New York, NY: Academic Press.

Dr. Beth Adelson
Department of Psychology
Rutgers University
Camden, NJ 08102

Technical Document Center
AL/HGR-TDC
Wright-Patterson AFB
OH 45433-6503

Dr. David L. Alderton, Code 131
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. James A. Anderson
Department of Cognitive and
Linguistic Sciences
Brown University
Box 1978
Providence, RI 02912

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

Dr. Stephen J. Andriole, Chairman
College of Information Studies
Drexel University
Philadelphia, PA 19104

Dr. Gregory Anrig
Educational Testing Service
Princeton, NJ 08541

Technical Director, ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Edward Atkins
13705 Lakewood Ct.
Rockville, MD 20850

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

Dr. Patricia Baggett
School of Education
610 E. University, Rm 1302D
University of Michigan
Ann Arbor, MI 48109-1259

Dr. Meryl S. Baker
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Pierre Baldi
Division of Biology, 216-76
California Institute of Technology
Pasadena, CA 91125

Dr. William O. Berry
Director of Life and
Environmental Sciences
AFOSR/NL, N1, Bldg. 410
Bolling AFB, DC 20332-6448

Dr. Menucha Birenbaum
Educational Testing
Service
Princeton, NJ 08541

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

Dr. J. C. Boudreaux
Manufacturing Engineering Lab
National Institute of
Standards and Technology
Gaithersburg, MD 20899

Dr. Gordon H. Bower
Department of Psychology

Stanford University
Stanford, CA 94306

Dr. Robert Breaux
Code 252
Naval Training Systems Center
Orlando, FL 32826-3224

Dr. Ann Brown
Graduate School of Education
University of California
EMST-4533 Tolman Hall
Berkeley, CA 94720

CDR Robert Carter
HCI Laboratory
Naval Research Laboratory
Washington, DC 20090

Dr. Eduardo Cascallar
Educational Testing Service
Rosedale Road
Princeton, NJ 08541

Dr. Paul R. Chatelier
Perceptronics
1911 North Ft. Myer Dr.
Suite 800
Arlington, VA 22209

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

Dr. Susan Chipman
Cognitive Science Program
Office of Naval Research
800 North Quincy St.
Arlington, VA 22217-5000

Dr. Raymond E. Christal
UES LAMP Science Advisor
AL/HRMIL
Brooks AFB, TX 78235

Dr. David E. Clement
Department of Psychology
University of South Carolina
Columbia, SC 29208

Dr. Rodney Cocking
NIMH, Basic Behavior and
Cognitive Science Research
5600 Fishers Lane, Rm 11C-10
Parklawn Building
Rockville, MD 20857

Director, Life Sciences
Office of Naval Research
Code 114
Arlington, VA 22217-5000

Director, Cognitive and
Neural Sciences, Code 1142
Office of Naval Research
Arlington, VA 22217-5000

Director
Training Systems Department
Code 15A
Navy Personnel R&D Center
San Diego, CA 92152-6800

Library, Code 231
Navy Personnel R&D Center
San Diego, CA 92152-5800

Commanding Officer
Naval Research Laboratory
Code 4827
Washington, DC 20375-5000

Dr. Albert T. Corbett
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Michael Cowen
Code 142
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Kenneth B. Cross
Anacapa Sciences, Inc.
P.O. Box 519
Santa Barbara, CA 93102

Dr. Peter Culicover
Director, Center for Cognitive
Science
208 Ohio Stadium East
1961 Tuttle Park Place
Columbus, OH 43210-1102

Dr. Diane Damos
Dept. of Human Factors
Institute of Safety &
Systems Management
Univ. of Southern California
Los Angeles, CA 90089-0021

Margaret Day, Librarian
Applied Science Associates
P.O. Box 1072
Butler, PA 16003

Dr. Geory Delacote
Exploratorium
3601 Lyon Street
San Francisco, CA
94123

Dr. Sharon Derry
Florida State University
Department of Psychology
Tallahassee, FL 32306

Dr. Stephanie Doane
University of Illinois
Department of Psychology
603 East Daniel Street
Champaign, IL 61820

Dr. Emanuel Donchin
University of Illinois
Department of Psychology
603 East Daniel Street
Champaign, IL 61820

Defense Technical
Information Center
DTIC/DDA-2
Cameron Station, Bldg 5
Alexandria, VA 22314
(4 Copies)

Dr. Richard Duran
Graduate School of Education
University of California
Santa Barbara, CA 93106

Dr. John Ellis
Navy Personnel R&D Center
Code 15
San Diego, CA 92152-6800

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

ERIC Facility-Acquisitions
1301 Piccard Drive, Suite 300
Rockville, MD 20850-4305

Dr. K. Anders Ericsson
University of Colorado
Department of Psychology
Campus Box 345
Boulder, CO 80309-0345

Dr. Martha Evens
Dept. of Computer Science
Illinois Institute of Technology
10 West 31st Street
Chicago, IL 60616

Dr. Beatrice J. Farr
Army Research Institute

COPY AVAILABLE TO DTIC DOES NOT PERMIT FULLY LEGIBLE REPRODUCTION

PERI-IC
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Marshall J. Farr
Farr-Sight Co.
2520 North Vernon Street
Arlington, VA 22207

Dr. Kathleen Fernandes
Naval Ocean Systems Center
Code 423
San Diego, CA 92152

Mr. Wallace Feurzeig
Educational Technology
Bolt Beranek & Newman
10 Moulton St.
Cambridge, MA 02238

Dr. Victor Fields
Department of Psychology
Montgomery College
Rockville, MD 20850

Dr. J. D. Fletcher
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311

Dr. Lawrence T. Frase
Executive Director
Division of Cognitive and
Instructional Science
Educational Testing Service
Princeton, NJ 08541

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

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

Dr. Alfred R. Fregly
AFOSR/NL, Bldg. 410
Bolling AFB, DC 20332-6448

Dr. Alinda Friedman
Department of Psychology
University of Alberta
Edmonton, Alberta
CANADA T6G 2E9

Dr. Jack J. Gelfand
Department of Psychology
Princeton University
Princeton, NJ 08544-1010

Dr. Dedre Gentner
Northwestern University
Department of Psychology
2029 Sheridan Road
Swift Hall, Rm 102
Evanston, IL 60208-2710

Chair, Department of
Computer Science
George Mason University
Fairfax, VA 22030

Dr. Helen Gigley
Naval Research Lab., Code 5530
4555 Overlook Avenue, S. W.
Washington, DC 20375-5000

Dr. Philip Gillis
ARI Fort Gordon
ATTN: PERI-ICD
Fort Gordon, GA 30905

Mr. Lee Gladwin
305 Davis Avenue
Leesburg, VA 22075

Dr. Robert Glaser
Learning Research
& Development Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

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

Dr. Sam Glucksberg
Department of Psychology
Princeton University
Princeton, NJ 08544-1010

Dr. Susan R. Goldman
Peabody College, Box 45
Vanderbilt University
Nashville, TN 37203

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

Dr. Arthur C. Graesser
Department of Psychology
Memphis State University
Memphis, TN 38152

Dr. Wayne Gray
Graduate School of Education
Fordham University
113 West 60th Street
New York, NY 10023

Dr. James G. Greeno
School of Education
Stanford University
Room 311
Stanford, CA 94305

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

Dr. Michael Habon
DORNIER GMBH
P.O. Box 1420
D-7990 Friedrichshafen 1
WEST GERMANY

Marilyn Halpern, Librarian
Brigham Library
Educational Testing Service
Carter and Rosedale Roads
Princeton, NJ 08541

Dr. Kristian J. Hammond
Department of Computer Science
University of Chicago
1100 E. 58th St.
Chicago, IL 60637

Dr. Stephen J. Hanson
Learning & Knowledge
Acquisition Research
Siemens Research Center
755 College Road East
Princeton, NJ 08540

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

Dr. Keith Holyoak
Department of Psychology
University of California
Los Angeles, CA 90024

Ms. Julia S. Hough
Cambridge University Press
40 West 20th Street
New York, NY 10011

Dr. William Howell
Chief Scientist
AFHRL/CA
Brooks AFB, TX 78235-5601

Dr. Earl Hunt
Dept. of Psychology, NI-25
University of Washington
Seattle, WA 98195

Dr. Jack Hunter
2122 Coolidge Street
Lansing, MI 48906

Dr. Martin J. Ippel
Center for the Study of
Education and Instruction
Leiden University
P. O. Box 9555
2300 RB Leiden
THE NETHERLANDS

Z. Jacobson
Bureau of Management Consulting
701-365 Laurier Ave., W.
Ottawa, Ontario K1H 5W3
CANADA

Dr. Robert Jannarone
Elec. and Computer Eng. Dept.
University of South Carolina
Columbia, SC 29208

Dr. Claude Janvier
CIRADE, X-7120
UQAM
P. O. Box 8888, Succ. A
Montreal, Quebec H3C 3P8
CANADA

Dr. Robin Jeffries
Hewlett-Packard
Laboratories, 1U-17
P.O. Box 10490
Palo Alto, CA 94303-0969

Dr. Edgar M. Johnson
Technical Director
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

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

Dr. Milton S. Katz
PSC 802 Box 15
FPO AE 09499-1500

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

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

Dr. David Kieras
Technical Communication Program
TIDAL Bldg., 2360 Bonisteel Blvd.
University of Michigan
Ann Arbor, MI 48109-2108

Dr. Sung Ho Kim
Educational Testing Service

Princeton, NJ 08541

Dr. Sylvan Kornblum
University of Michigan
Mental Health Research Institute
205 Washtenaw Place
Ann Arbor, MI 48109

Dr. Richard J. Koubek
School of Industrial
Engineering
Grissom Hall
Purdue University
West Lafayette, IN 47907

Dr. Benjamin Kuipers
University of Texas at Austin
~~Department of Computer Sciences~~
Austin, Texas 78712

Dr. Patrick Kyllonen
AFHRL/MOEL
Brooks AFB, TX 78235

Dr. Anita Lancaster
Assistant Director
Defense Manpower Data Center
Fourth Floor
1600 Wilson Blvd.
Arlington, VA 22209

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

Richard Lanterman
Commandant (G-PWP)
US Coast Guard
2100 Second St., SW
Washington, DC 20593-0001

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

Dr. Paul E. Lehner
Department of Information
Systems & Engineering
George Mason University
4400 University Drive
Fairfax, VA 22030-4444

Dr. Alan M. Lesgold
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Richard Lesh
Educational Testing Service
Princeton, NJ 08541

Dr. Michael Levine
Educational Psychology
210 Education Bldg.
1310 South Sixth Street
University of IL at
Urbana-Champaign
Champaign, IL 61820-6990

Dr. Marcia C. Linn
Graduate School
of Education, EMST
Tolman Hall
University of California
Berkeley, CA 94720

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

Dr. Evans Mandes
George Mason University
4400 University Drive
Fairfax, VA 22030

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

Dr. Clessen J. Martin
Head, DoD Coordinator,
Recruiting Plans and Programs
Branch Code: PERS 2FF/234
Navy Annex, Room 2832
Washington, DC 20350

Dr. Elizabeth Martin
AL/HRA, Stop 44
Williams AFB
AZ 85240

Dr. John Mayer
Technical Communications
Program
2360 Bonisteel Blvd.
University of Michigan
Ann Arbor, MI 48109

Dr. Richard E. Mayer
Department of Psychology
University of California
Santa Barbara, CA 93106

Dr. Kathleen McKeown
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027

Dr. Joseph McLachlan
Navy Personnel Research
and Development Center
Code 14
San Diego, CA 92152-6800

Dr. Michael McNeese
DET-1, AL/CFHI
BLDG 248
Wright-Patterson AFB, OH 45432

Dr. Alan Meyrowitz
Naval Research Laboratory
Code 5510
4555 Overlook Ave., SW
Washington, DC 20375-5000

Dr. Ryszard S. Michalski
Center for Artificial Intelligence
George Mason University
Science and Tech II, Rm.411
4400 University Drive
Fairfax, VA 22030-4444

Dr. Andrew R. Molnar
Applications of Advanced
Technologies, Rm. 635
National Science Foundation
Washington, DC 20550

Dr. Randy Mumaw
Human Sciences
Westinghouse Science
& Technology Ctr.
1310 Beulah Road
Pittsburgh, PA 15235

Dr. Allen Munro
Behavioral Technology
Laboratories - USC
250 N. Harbor Dr., Suite 309
Redondo Beach, CA 90277

Distribution Support Division
Bureau of Naval Personnel
PERS-471D
Washington, DC 20370

Academic Progs. & Research Branch
Naval Technical Training Command
Code N-62
NAS Memphis (75)
Millington, TN 30854

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

Dr. Donald A. Norman
Department of Cognitive Science
University of California
La Jolla, CA 92093-0515

Director
Training Systems Department
NPRDC (Code 14)
San Diego, CA 92152-6800

Library, NPRDC
Code 041
San Diego, CA 92152-6800

Librarian
Naval Center for Applied Research
in Artificial Intelligence
Naval Research Laboratory
Code 5510
Washington, DC 20375-5000

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

Mathematics Division
Office of Naval Research
Code 1111
800 North Quincy Street
Arlington, VA 22217-5000

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

Assistant for Training Technology
and Human Factors
Office of the DCNO(MPT) (Op-11E)
Department of the Navy
Washington, DC 20350-2000

Special Assistant for Research
Management
Chief of Naval Personnel (PERS-01JT)
Department of the Navy
Washington, DC 20350-2000

Dr. Judith Orasanu
Mail Stop 239-1
NASA Ames Research Center
Moffett Field, CA 94035

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

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

Dr. Roy Pea
Institute for the
Learning Sciences
Northwestern University
1890 Maple Avenue
Evanston, IL 60201

Dr. Ray S. Perez
ARI (PERI-II)

5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Nancy N. Perry
Naval Education and Training
Program Support Activity
Code-047
Building 2435
Pensacola, FL 32509-5000

CDR Frank C. Petho
Naval Postgraduate
School
Code OR/PE
Monterey, CA 93943

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

Dr. Elizabeth A. Phelps
Department of Psychology
New School for Social Research
65 Fifth Avenue
New York, NY 10003

Dr. Peter Pirolli
School of Education
University of California
Berkeley, CA 94720

Dr. Martha Polson
Department of Psychology
University of Colorado
Boulder, CO 80309-0344

Dr. Peter Polson
University of Colorado
Department of Psychology
Boulder, CO 80309-0344

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

Psyc Info - CD and M
American Psychological Assoc.
1200 Uhle Street
Arlington, VA 22201

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

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

Dr. Fred Reif
CDEC, Smith Hall
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Charles M. Reigeluth
Chairman, Instructional
Systems Technology
School of Education, Rm. 210
Indiana University
Bloomington, IN 47405

Dr. Daniel Reisberg
Reed College
Department of Psychology
Portland, OR 97202

Dr. Brian Reiser
Institute for the Learning Sciences
Northwestern University
1890 Maple Avenue
Evanston, IL 60201-3142

Dr. Lauren Resnick
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15213

Mr. W. A. Rizzo
Head, Human Factors Division
Naval Training Systems Center
Code 26
12350 Research Parkway
Orlando, FL 32826-3224

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

Dr. Andrew M. Rose
American Institutes
for Research
3333 K Street, N.W.
Washington, DC 20007

Dr. Eduardo Salas
Human Factors Division (Code 262)
12350 Research Parkway
Naval Training Systems Center
Orlando, FL 32826-3224

Dr. Mark Schlager
SRI International
333 Ravenswood Ave.
Room BS-131
Menlo Park, CA 94025

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

Dr. Alan H. Schoenfeld
University of California
Department of Education
Berkeley, CA 94720

Dr. Mary Schratz
4100 Parkside
Carlsbad, CA 92008

Dr. Judith W. Segal
OERI
555 New Jersey Ave., NW
Washington, DC 20208-5648

Dr. Robert J. Seidel
US Army Research Institute
5001 Eisenhower Ave.
Alexandria, VA 22333

Dr. Colleen M. Seifert
Department of Psychology
University of Michigan
330 Packard Road
Ann Arbor, MI 48104

Dr. Terrence J. Sejnowski
Professor
The Salk Institute
P. O. Box 85800
San Diego, CA 92138-9216

Dr. Michael G. Shafto
NASA Ames Research Ctr.
Mail Stop 262-1
Moffett Field, CA 94035-1000

Mr. Richard J. Shavelson
Graduate School of Education
University of California
Santa Barbara, CA 93106

Dr. Jerome E. Singer
Department of Medical Psychology
Uniformed Services Univ. of the
Health Sciences
4301 Jones Bridge Road
Bethesda, MD 20814-4799

Dr. Jan Sinnott

Department of
Computer Science
Towson State University
Towson, MD 21204

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

Dr. Robert Smillie
Naval Ocean Systems Center
Code 443
San Diego, CA 92152-5000

Dr. Richard E. Snow
School of Education
Stanford University
Stanford, CA 94305

Dr. Kathryn T. Spoehr
Department of Cognitive
& Linguistic Sciences
Box 1978
Brown University
Providence, RI 02912

Dr. James J. Staszewski
Dept. of Psychology
University of South Carolina
Columbia, SC 29210

Dean, College of Behavioral
and Social Sciences
University of Maryland,
Baltimore County
Baltimore, MD 21228

Dr. Kurt VanLehn
Learning Research
& Development Ctr.
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Jerry Vogt
Department of Psychology
St. Norbert College
De Pere, WI 54115-2099

Dr. Douglas Wetzel
Code 15
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Barbara White
School of Education
Tolman Hall, EMST
University of California
Berkeley, CA 94720

Alexandra K. Wigdor
NRC MH-176
2101 Constitution Ave.
Washington, DC 20418

Dr. David Wiley
School of Education
and Social Policy
Northwestern University
Evanston, IL 60208

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

Dr. Bruce Williams
Department of Educational
Psychology
University of Illinois
Urbana, IL 61801

Dr. Kent E. Williams
Virginia Polytechnic Institute
and State University

Department of Industrial and
Systems Engineering/MSL
1900 Kraft Drive
Blacksburg, VA 24060

Dr. Robert A. Wisher
U.S. Army Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Merlin C. Wittrock
Graduate School of Education
Univ. of Calif., Los Angeles
Los Angeles, CA 90024

Dr. Kentaro Yamamoto
03-0T
Educational Testing Service
Rosedale Road
Princeton, NJ 08541

Frank R. Yekovich
Dept. of Education
Catholic University
Washington, DC 20064

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

FOUNDATIONS OF KNOWLEDGE ACQUISITION: Cognitive Models of Complex Learning

edited
by

Susan Chipman
Office of Naval Research

Alan L. Meyrowitz
Naval Research Laboratory

KLUWER ACADEMIC PUBLISHERS
Boston/Dordrecht/London

Distributors for North America:
Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061 USA

Distributors for all other countries:
Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS

Library of Congress Cataloging-in-Publication Data

Foundations of knowledge acquisition / edited by Susan Chipman, Alan
L. Meyrowitz
p. cm. -- (The Kluwer international series in engineering and
computer science ; SECS 194)
Includes bibliographical references and index.
Contents: v. [1] Cognitive models of complex learning.
ISBN 0-7923-9277-9
1. Knowledge acquisition (Expert systems) I. Chipman, Susan.
II. Meyrowitz, Alan Lester. III. Series.
QA76.76.E95F68 1993
006.3'1--dc20 92-36720
CIP

Printed on acid-free paper.

Printed in the United States of America