Cognitive Skill Acquisition

Kurt VanLehn
Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA 15260
VanLehn@cs.pitt.edu
(412)624-7458 fax (412) 624-9149

May 4, 1995

Abstract

Cognitive skills acquisition is acquiring the ability to solve problems in intellectual tasks, where success is determined more by the subjects' knowledge than their physical prowess. This chapter reviews reseach conducted in the last ten years on cognitive skill acquisition. It covers the initial stages of acquiring a single principle or rule, the initial stages of acquiring a collection of interacting pieces of knowledge, and the final stages of acquiring a skill, wherein practice causes increases speed and accuracy.

To appear in *Annual Review of Psychology, Vol. 47*, J. Spence, J. Darly & D. J. Foss (Eds). Palo Alto, CA: Annual Reviews.

Keywords: Cognitive skill acquisition, problem solving, schema acquisition, practice effects, transfer, self-explanation

Short title: Cognitive skill acquisition.

Contents

1 Introduction

Cognitive skills acquisition is acquiring the ability to solve problems in intellectual tasks, where success is determined more by the subjects' knowledge than their physical prowess. Frequently studied tasks include solving algebraic equations and word problems, college physics problem solving, computer programming, medical diagnosis and electronic troubleshooting. Researchers in cognitive skill acquisition study how people learn to accomplish such complex, knowledge-intensive tasks and how they become experts in their fields.

1.1 A brief history

Cognitive skill acquisition has its historical roots in the study of problem solving. Research on problem solving began early in this century by studying what makes problems difficult to solve (Duncan, 1959). In the 1960s, researchers turned to studying the process of solving a problem. Subjects solved multistep puzzles while explaining their reasoning aloud. Transcriptions of their commentaries, called *verbal protocols*, provided the empirical foundations for developing computational models of problem solving. Newell and Simon (1972) introduced most of the important theoretical concepts, including problem spaces, search trees and production systems.

Because problem solving research emphasized the process of moving from one intermediate state to another until one finally arrived at a solution, researchers preferred to use tasks where most intermediate states were physical states. In the Tower of Hanoi, for instance, the subjects try to move a pyramidal stack of disks from one peg to another by moving one disk at a time subject to certain restrictions. Solving the puzzle requires many physical movements of disks and thus exposes the subjects' intermediate states. Problems that are solved with a single physical action were seldom studied by problem solving researchers.

During the 1960s, two related fields developed. *Decision making* studied people making a choice under uncertainty, and *reasoning* studied people drawing a conclusion from a combination of mental inferences. In a sense, these are also forms of problem solving. However, perhaps because most of their intermediate states are mental and not physical, their methodological and theoretical concerns have remained distinct from those of problem solving.

In the 1970's, problem solving researchers became interested in how subjects solved problems that required much more knowledge than the simple puzzle problems that were used in the 1960's. They studied problems in chess, physics, mathematics, computer programming, medical diagnosis and many other fields. Whereas one could tell a subject everything they needed to know to solve a puzzle in a few minutes, solution of even an easy problem in a knowledge-rich task domain required many of hours of preparatory training.

The exploration of knowledge-rich problem solving began by contrasting the performance of experts and novices. For instance, one robust finding is that experts can sort problems into categories according to features of their solutions, while novices can only sort problems using features of the problem statement itself. As discussed by Ericsson & Lehmann (this volume), this and many other the findings can be explained by assuming that whenever mental planning of solutions is possible (e.g., because all the information required to solve problems is present in the initial state), then experts typically develop the ability to plan solutions in memory. This often requires the ability to envision sequences of intermediate states, so experts develop impressive mneumonic powers, but only for intermediate states that they typically encounter.

In the 1980's, many researchers turned to studying how people acquired expertise. Attention initially focused on the role of practice in the development of expertise. Phenomena that were often found with motor skills, such as the power-law of practice and the identical elements model of transfer, were found with cognitive skills as well. Most of the recent work has focussed on the role of instruction during the early stages of skill acquisition, and in particular, on the role of examples. In this literature, an *example* is a problem whose solution is given to the student, along with the solution's derivation. Examples appear to play a central role in the early phases of cognitive skill acquisition.

Because reviews of the early days of problem solving and cognitive skill acquisition are available (VanLehn, 1989; Kahney, 1993), as well as reviews of expertise (Ericsson & Lehmann, this volume) and instructional considerations (Glaser & Bassock, 1989; Voss, Wiley & Carretero, 1995) , this review will focus exclusively on recent work in cognitive skill acquisition. Because there is much material to cover and simply listing the major findings would make it impossible to assimilate them all, a loose framework has been provided.

1.2 A framework for reviewing cognitive skill acquisition

Fitts (1964) distinguished three phases of motor skill acquisition. His *early*, *intermediate* and *late* phases also aptly describe the course of cognitive skill acquisition.

During the early phase, the subject is trying to understand the domain knowledge without yet trying to apply it. This phase is dominated by reading, discussion and other general-purpose information acquisition activities that lie outside the scope of this review. Most investigations of cognitive skill acquisition do not collect observations during the early phase.

The intermediate phase begins when students turn their attention to solving problems. Before they begin solving problems themselves, they often study a few problems that have already been solved (called *examples* henceforth). Examples may be printed in a textbook or be presented "live" by a teacher. As they solve problems, students may refer back to the textbook or ask a teacher for help, but their primary focus is on solving problems. This differentiates the intermediate phase from the early phase, where the primary focus is on studying expository instructional material.

When subjects start the intermediate phase, they have some relevant knowledge for solving problems but certainly not all of it. They also may have acquired some misunderstandings. Thus, the first order of business is to correct these flaws in the domain knowledge. (For lack of a better word, *flaw* will be used subsequently to stand both for missing knowledge and incorrect knowledge.) The second order of business is to acquire heuristic, experiential knowledge that expedites problem solving.

Eventually, students remove all the flaws in their knowledge and can solve problems without conceptual errors, although they may still make unintended errors, or "slips" (Norman, 1982). This capability signals the end of the intermediate phase and the beginning of the late phase. During the late phase, students continue to improve in speed and accuracy as they practice even though their understanding of the domain and their basic approach to solving problems does not change. Practice effects and transfer are the main research issues here.

This 3-phase chronology is an idealization. The boundaries between phases are not as sharp as the description above would lead one to believe. Moreover, instruction on a cognitive skill is divided into courses, topics, chapters and sections. Students are introduced to a component of the skill, given substantial practice with it, then moved on to the next component. Thus, at any given time, students may be in the late phase with respect to some components of their skill, but in other phases with respect to other components. Nonetheless, it is useful to make the 3-phase distinction, because different empirical phenomena characterize each phase.

This review covers the intermediate and final phases. However, much more work has been done on the intermediate phase than the final phase, so the review of intermediate phase research is split into two parts. The first part covers studies where students learned only a single principle. These studies focussed on the basic processes of assimilating a new principle, retrieving it from memory during problem solving, and applying it. The second part covers studies where students learn many principles. Following this two-part discussion of the intermediate phase is a section on the late phase.

2 The intermediate phase: Learning a single principle

Before beginning the discussion of learning a single principle, some general remarks about the intermediate phase are necessary.

Perhaps the most ubiquitous finding of the intermediate phase is the importance of examples. Studies of students in a variety of instructional situations have found that students prefer learning from examples than learning from other forms of instruction (e.g., Lefevre & Dixon, 1986; Pirolli & Anderson, 1985; Chi et al, 1989). Students learn more from studying examples than from solving the same problems themselves (Cooper & Sweller, 1987; Carroll, 1994). A 3-year program in algebra was completed in 2 years by students who only studied examples and solved problems without lectures or other direct instruction (Zhu & Simon, 1987). Because of the importance of examples, most research on the intermediate phase has used instructional material where examples are prominent. Sometimes, the instruction consists only of examples, and students must infer the general principles themselves.

Most research has studied students solving problems alone. There is some research on learning from a tutor, which is covered at the end of the intermediate phase section.

Learning by solving problems in small groups has not yet been studied much by cognitive skill acquisition researchers.

2.1 Learning a single principle: Introduction

As mentioned earlier, much work on the intermediate phase has concentrated on learning material that is about the size of a single principle, where a principle is the sort of thing that a textbook states in a colored box and discusses for several pages. Often the instruction includes an example illustrating the application of the principle. Table 1 shows an example, taken from Catrambone (1994), illustrating an elementary probability principle that is often used in cognitive skill acquisition research. Knowledge of the principle includes not only the permutation formula, but also the meaning of the variables n and r and the kinds of problems this formula applies to. The convergence-of-forces idea used to solve Duncker's famous X-ray problem is another commonly used principle. One can generally teach a subject a rudimentary version of a principle in an hour or less, in contrast to teaching a rudimentary version of a whole cognitive skill, which might take days or months.

All the experiments to be discussed have a similar format. Subjects are trained, typically by studying a booklet. The training material almost always includes examples, and may consist only of examples. After the training (and sometimes after a distractor task), the subjects are given problems to solve. The problems can be easily solved using the principle but are difficult to solve without it.

When the training consists only of examples, then solving the test problems is sometimes called *analogical problem solving* because it involves finding an analogy (correspondence) between the example and the problem.

Although learning a principle consists of early, intermediate and late phases, these experiments focus exclusively on the intermediate phase. Students are not given enough practice to enter the late phase. The early phase in these experiments consists of studying the booklet or other instructional material, and observations are rarely collected during this phase.

The intermediate phase consists of trying to solve problems by applying the new principle, example or both. Applying a principle or example consists of retrieving it, placing

Table 1: An example solved with the permutation principle

Problem: The supply department at IBM has to make sure that scientists get computers. Today, they have 11 IBM computers and 8 IBM scientists requesting computers. The scientists randomly choose their computer, but do so in alphabetical order. What is the probability that the first 3 scientists alphabetically will get the lowest, second lowest, and third lowest serial numbers, respectively, on their computers?

Solution: The equation needed for this problem is

$$\frac{1}{n*(n-1)*\ldots*(n-r+1)}.$$

This equation allows one to determine the probability of the above outcome occurring. In this problem, $n = 11$ and $r = 3$. The 11 represents the number of computers that are available to be chosen while the 3 represents the number of choices that are being focused on in this problem. The equation divides the number of ways the desired outcome could occur by the number of possible outcomes. So, inserting 11 and 3 into the equation, we find that the overall probability is

$$\frac{1}{11*10*9} = \frac{1}{990}$$

its parts into correspondence with parts of the problem (e.g., in the case of the permutation principle, deciding what n and r are), and drawing inferences about the problem and its solution on the basis of the problem's correspondence with the principle or example. After applying the principle or example, subjects may generalize it. Each of these processes—retrieval, mapping, application and generalization—will be discussed in turn.

2.2 Retrieval

There appear to be two kinds of retrieval: spontaneous and deliberate. Deliberate retrieval often occurs after subjects are given a hint (e.g., "The examples you studied earlier will help you solve this problem."), or when the experiment simulates an instructional situation where students expect earlier material to be relevant to solving problems (Brown & Kane, 1988). Spontaneous retrieval, or *reminding* as it is more commonly known, occurs when the experimenter hides the relationship between the training and testing phases of the experiment (typically by telling students that they are two different experiments), and yet subjects nonetheless notice that the training is relevant to solving the problem. This experimental paradigm explores why education so often creates "inert" knowledge, which students can recall when given explicit cues but fail to apply outside the classroom (Bransford et al, 1989).

A strong but obvious effect is that deliberate retrieval is vastly more successful than reminding. Many more subjects can retrieve a principle or example after a hint than before a hint (e.g., Gick & Holyoak, 1980).

When reminding does occur, it often is triggered by surface similarities between the problem and a training example (e.g., Catrambone & Holyoak, 1989; Holyoak & Koh, 1987; Ross, 1984, 1987, 1989). For instance, subjects are likely to be reminded of the IBM example (Table 1) more by a problem that mentions Microsoft programmers designing computer software than by the car mechanics problem of Table 2 even though the car mechanics problem has same mathematical structure as the IBM example. Subjects can be reminded by structural similarities, but only when the training emphasizes the underlying structure of the examples, and the test problems are reworded to emphasize their deep structure (Catrambone & Holyoak, 1989).

Table 2: A problem with the same deep structure as the IBM problem

Southside High School has a vocation car mechanics class in which students repair cars. One day there are 15 students and 18 cars requiring repairs. The cars are assigned to students in order of the severity of their damages (the car in the worst shape goes first), but the student to work on the car is randomly chosen. What is the probability that the 6 cars in the worst shape are worked on by the 6 students with the highest grades, in order of their grades (i.e., the student with the highest grade working on the worst car, etc.)?

Although superficial reminding seems to be the population norm, it is less common among students with high mathematics SAT scores (Novick & Holyoak, 1991; Novick, 1988). Moreover, deliberate retrieval, particularly in instructional situations, is often guided by structural similarity (Faries & Reiser, 1988).

2.3 Mapping

The mapping process puts parts of the principle or example into correspondence with parts of the problem. For instance, in order to use the example of Table 1 as a model for solving a new problem, subjects must find values for n and r, which in turn requires finding objects corresponding to the IBM scientists and computers.

Subjects are easily misled by surface similarities into using the wrong correspondence (mapping). For instance, Ross (1989) found that most subjects solved the problem of Table 2 by pairing mechanics with IBM scientists and cars with computers. This matches the superficial characteristics of the objects, but it produces an incorrect solution to the problem.

2.4 Application

In some cases, subjects have almost finished solving the problem after they have put the principle or example into correspondence with the problem. For instance, once subjects have mapped the car mechanics problem to the IBM example, the only remaining tasks are substituting 15 for n and 6 for r in the permutation formula and doing the arithmetic.

However, applying the principle or example can be much more involved (Novick & Holyoak, 1991). For instance, when an example is complex, subjects usually refer back to it many times when solving a problem (e.g., Pirolli & Anderson, 1985; Reed, Willis & Guarino, 1994; VanLehn, 1995a). They appear to use a variety of strategies for deciding whether to refer to the example and to attempt the next step without help (VanLehn, 1995a).

2.5 Generalization

Problems and examples contain information that is not causally or logically related to their solution. In the IBM example, the nature of the particular objects being chosen (IBM computers) is irrelevant. Such information comprises the surface features mentioned earlier. As discussed above, surface features can play a strong role in retrieval and mapping. Their role in application has not yet been established, but is likely to be strong as well.

Generalization is the process of modifying one's understanding of an example or principle in such a way that surface information does not play a role in retrieval, mapping and application. Generalization allows one to apply the principle or example to more problems.

A variety of instructional methods for encouraging generalization have been tried. Gick, Holyoak and others have found that simply augmenting the example with an explanation of the principle behind it provided little generalization. Using two examples was also ineffective. What does work is using two examples and some sort of "highlighting device" that encourages subjects to compare the examples and find their common structure (Gick & Holyoak, 1983; Catrambone & Holyoak, 1989; Ross & Kennedy, 1990; Cummins, 1992). Generalization causes subjects to be reminded of examples by structural features instead of just surface features, and to be fooled less during mapping by surface features.

These results strongly suggest that generalization is not an automatic process, as assumed by earlier theories of cognitive skill acquisition (e.g., Anderson, 1983). Although protocol-taking experiments would be necessary to confirm this suggestion, it is likely that subjects must actively decide which propositions in an example are structural and which are superficial. Alternatively, subjects can probably be told which aspects of an example are general (cf. Ahn, Brewer & Mooney, 1992).

As one would predict from basic properties of memory, deciding that some propositions in an example are superficial does not erase them from memory, nor does building a generalization from comparison of two examples erase the examples from memory (Bernardo, 1994; Novick & Holyoak, 1991).

2.6 Summary

This concludes the discussion of the learning of individual principles. Overall, superficial reasoning is the norm, but subjects can be induced to use deeper reasoning under certain circumstances. Left to their own devices, subjects typically just encode examples and principles in memory, where they languish until retrieved deliberately or via spontaneous, superficial association. Once retrieved, an example or principle is applied by just "plugging in" superficially similar objects and "copying" the resulting solutions. Subjects can use more structural features in spontaneous retrieval and mapping, but they must first be induced to generalize the examples, which seems to require using multiple examples and directing the subjects to find their commonalities.

On the other hand, when subjects suspect or are told that examples they have seen might help them solve their problem, their retrieval is often based on structural features. That is, they search memory or the textbook for examples whose solution might help them.

3 The intermediate phase: Learning multiple principles

Mastering a cognitive skill often requires learning more than one principle, as well as many other pieces of knowledge that one would hesitate to call "principles." For instance, a physics student must learn facts, such as the units for measuring force (Newtons) and mass (grams). There are even borderline cases: Is the knowledge that mass is not the same thing as weight a principle? Learning a cognitive skill also requires learning heuristics or experiences that will help one select the right combination of principles for solving a problem.

Much of what has been observed in the study of single-principle learning probably applies to the learning of all these types of knowledge. The processes of retrieval (both spontaneous and deliberate), mapping, application and generalization probably characterize the acquisition of minor principles (e.g., "mass and weight are different"), heuristics and other generalizations. Factual knowledge may be too simple for mapping and generalization

Table 3: Examples and problems of solving equations for a

$$ab + f = g \Longrightarrow ab = g - f \Longrightarrow a = (g - f)/b$$

$$b(a + f) = g \Longrightarrow a + f = g/b \Longrightarrow a = (g/b) - f$$

$$m(ac + b) = k$$

$$f(a + b) + w = g$$

to apply.

However, as the quantity and complexity of the material to be learned increases, one can expect to see effects that cannot be observed when studying the acquisition of a single principles. This section focuses on reviewing such phenomena.

3.1 Transfer

When an example uses multiple principles in its solution, then it is possible to study an interesting type of transfer. First students are trained with examples and problems that use two or more principles in a certain combination, then they are tested on problems that require using the principles in a different combination. For instance, consider the two algebra examples shown on the top two line of Table 3. Their solutions require using two principles: removing a term by subtracting it from both sides of an equation, and removing a coefficient by dividing both sides of the equation by it. These same principles can be used to solve the problems shown on the last two lines of Table 3, but they must be used in a different combination than they were used in the examples. If a student trained on the two examples can solve the two problems, then a certain rather constrained type of transfer has been obtained.

However, such transfer is rarely obtained. Cooper and Sweller (1987) found that many eight-grade students trained with multiple versions of the two examples in Table 3 could not solve the problems shown on the last two lines of the table. On the other hand, the students had no difficulty solving problems, such as $ac + g = h$, that could be solved by "copying" the solution of the training examples. Similar lack of transfer has been found many times (Reed, Dempster & Ettinger, 1982; Sweller & Cooper, 1985; Catrambone, 1994).

Note that this kind of transfer is different from the generalization discussed in the preceding section, where copying the examples' solution was all that was expected of the subjects. Generalizing the examples shown in the top two lines of Table 3 will solve problems such as $ac + g = h$ but not the ones shown in the bottom of Table 3.

Catrambone (1994a, 1994b) showed that modifying the examples' solutions in order to highlight the application of each principle significantly increased transfer. Training that required students to draw contrasts among pairs of examples in order to see individual principle applications was less successful (Catrambone & Holyoak, 1990, 1987).

Even when principle applications are not highlighted, principle-based transfer can occur, but it requires a large number of examples and above-average students (Cooper & Sweller, 1987). This suggests that some students study examples differently than others, and that is the topic of the next section.

3.2 Strategy differences in learning from examples

Examples are commonly used in two ways. Students study examples before solving problems, or they refer back to examples when they are in the midst of solving a problem. For each of these two activities, it appears that learning is strongly affected by the students' strategies for studying examples and for referring back to examples. Let us first consider the activity of studying examples before solving problems.

Chi et al (1989) found that students who explained examples thoroughly to themselves learned much more than students who merely read the example through. Chi et al had 9 college students first study four introductory chapters from a college physics textbook until they could pass exams on each chapter. This insured that all students had the necessary pre-requisite knowledge for understanding the target chapter. The target chapter taught students about forces and Newtons' laws. The students read the target chapter, studied 3 examples, and solved 19 problems. Protocols were taken during the example studying and problem solving phase. Students were classified via a median split on their problem solving scores into Good learners and Poor learners. When protocols of the example studying phase were analyzed, it was found that the Good learners uttered more self-explanations than Poor learners, where a *self-explanation* is any inference about the example that goes beyond the information presented in the example. For instance, given the line $F_{ax} = -F_a \cos(30^\circ)$, a Good student might say, "So F_{ax} must be the leg of the right triangle, and F_a is the hypotenuse ...yep. What's that minus sign doing there?" Poor students often just read the line or paraphrased it, then went on to the next line.

Chi and Vanlehn (1991) analyzed the content of the students' self-explanations. The results suggested that there were two general sources for self-explanations. One was deduction from knowledge acquired earlier while reading the text part of the chapter, usually by simply applying a general principle to information in the current example statement. The second source was generalization and extension of the example statement. These inferences helped to fill gaps in the students' knowledge, most often by providing necessary technical details that were not discussed in the text.

VanLehn, Jones and Chi (1992) constructed a computer model of the example studying and problem solving process. In order to solve the problems correctly, the model required approximately 60 rules. The rules represented major principles, minor principles, facts and technical details (e.g., how to determine whether the sign of a vector's component is positive or negative). Less than half of the rules were mentioned in the textbook.[1] The other rules were learned by the model as it studied the examples, but only if it self-explained them. The model accounted for the aggregate results from the Chi et al (1989) study as well as the performance of individual subjects (Vanlehn & Jones, 1993). Error patterns and other analyses are also consistent with the hypothesis that most of the benefit from self-explanation comes from filling in gaps in the subjects' knowledge (VanLehn & Jones, 1995).

The self-explanation effect is quite general. Self-explaining examples in a variety of task domains improves learning (Pirolli & Bielaczyc, 1989; Pirolli & Recker, 1994; Fergusson-Hessler & de Jong, 1990; Lovett, 1992; Brown & Kane, 1988; Pressley et al, 1992). Self-explaining expository text (Chi et al, 1994) and hyper-text (Recker & Pirolli, 1995) instead of examples also enhances learning. Most importantly, students can be trained to self-explain, and this improves their learning dramatically (Chi et al, 1994; Bielaczyc, Pirolli

[1] Other investigators who have formalized textbook knowledge have also found that the textbook leaves out many crucial details (e.g., Psotka, Massey & Mutter, 1988, section 1).

& Brown, in press; Bielaczyc & Recker, 1991). However, self-explanation only affects the initial acquisition of knowledge and not subsequent improvement with practice (Pirolli & Recker, 1994).

Several of these studies also found a self-monitoring effect (Chi et al, 1989; Pirolli & Recker, 1994). Both Good and Poor students tended to spontaneously utter assessments of their understanding. However, Poor students tended to utter uniformly positive self-assessments (e.g., "Yep. That makes sense.") even though their subsequent performance indicates that they really did not understand the material. On the other hand, Good students tended to monitor their understanding more accurately, and frequently noted failures to understand (e.g., "Wait. I don't see how they got that."). Accuracy in self-monitoring appears to be correlated with learning.

So far, we have been discussing different strategies for studying examples before solving problems. However, students also refer to examples as they solve problems, and here we also find correlations between learning and strategies. Both Good and Poor learners refer to examples during problem solving (Chi et al, 1989), although more so with the first few problems solved than with later problems (Pirolli, 1991; Reed, Willis & Guarino, 1994). What matters is the way students refer to examples. Protocol analyses (Chi et al, 1989; VanLehn, 1995a,1995b) and latency analyses (Pirolli & Recker, 1994) suggest that Poor learners maximize their use of problem-solving by analogy: they refer to an example as soon as they notice that it is relevant and copy as much of its solution as possible. On the other hand, Good learners minimize their use of analogies to examples: they refer to the example only when they get stuck solving a problem and need some help. That is, Good solvers prefer to solve problems by themselves, whereas Poor solvers prefer to adapt an example's solution.

After solving a problem, subjects often reflect on the solution. Pirolli & Recker (1994) found that good and poor learners both reflect on half the problems they solve, but what they say is different. Poor learners usually just paraphrased the solution. Good learners tried to abstract general solution methods, often by comparing this problem's solution to the solutions of earlier problems.

In all these studies, the criteria for learning included performance on problems that could not be solved by merely copying the example solutions. Thus, learning requires the kind of transfer discussed earlier (section 3.1). Strategy effects may not show up on single-principle experiments (covered in section 2) because learning (generalization) is assessed on problems that can be solved by copying solutions.

In short, learners seem to use different studying strategies that trade off effort and liklihood of learning. Some students exert considerable time and energy in self-explaining examples and/or by trying to solve problems without copying the examples; these students often learn more. Other students exert less effort by merely reading the examples through and/or by copying example solutions; they learn less. As the saying goes: no pain, no gain.

3.3 Learning events

Knowledge of a complex skill is composed of many pieces of knowledge. Some represent principles, some represent examples or generalizations thereof, and others represent technical details, heuristics and other information that could be relevant in solving a problem. The studies reviewed earlier suggest that learning a single principle requires attention, both during the early phase (studying a booklet, typically) and during the intermediate phase as the principle is retrieved, mapped, applied and generalized. Moreover, it appears that generalization of an example requires attention—it is not an automatic process nor a mere by-product of applying the example to solve a problem. This in turn suggests that during the intermediate phase of complex skill acquisition, new pieces of domain knowledge are acquired one at a time, by taking time off from problem solving or example studying to attend to the discovery of new pieces of knowledge and perhaps to generalization as well.

This hypothesis is consistent with several studies of transfer. For instance, Bovair, Kieras and Polson (1990) taught subjects sequences of simple text-editing commands. Each command had several steps, such as pressing a function key (copy, delete, move, etc.), checking a prompt, selecting some text, and pressing the enter key. They represented each command's procedure with a set of rules. Across commands, some rules were identical, some were analogous, and some appeared in only one command. They trained subjects on all commands, varying the order in which the commands were taught. For each order, they calculated the number of new rules to be learned for each command, the number of rules that were identical to rules learned earlier, and the number of rules that were only analogous to rules learned earlier. They found that the training time was a linear function of the number of new rules and the total number rules. There was a 23 second cost for each rule, because the subject had to execute every rule of the command, regardless of whether that rule had already been learned or not. However, each new rule to be learned required a substantial additional amount of time, about 30 seconds. This suggests that learning a new rule requires taking time off from executing the familiar parts of a procedure. There was no cost in training time for learning rules that were analogous to rules learned earlier, which suggests generalization of these rules was quite easy. Singley and Anderson (1989) also found, using text editing, programming and mathematical task domains, that training time was a function of the number of new rules to be learned.

If learning a new principle or other piece of knowledge does take time away from executing familiar knowledge, then perhaps one will see *learning events* during problem solving, wherein a subjects briefly switches attention from solving the problem to reasoning about the domain knowledge itself. Learning events should also be found during other instructional activities, such as self-explanation. As it turns out, learning events can be observed via detailed protocol analysis.

Learning events were first observed in discovery learning situations, wherein students were given very little instruction and had to discover the principles of a domain during the course of solving problems in it. Karmiloff-Smith and Inhelder (1975) observed discrete changes in children's procedures for balancing blocks on a beam, and explained them as miniature discoveries—learning events. Kuhn and Phelps (1982) also noted discrete changes in college students' experiment design strategies, and assumed that they were due to learning events. Siegler and Jenkins (1989) observed children shift their strategies for adding by counting on their fingers, and coined the term "learning event" for the brief episodes where the changes occurred. VanLehn (1991) showed that all the changes in strategy of a subject solving the Tower of Hanoi puzzle seemed to appear during learning events.

Discovery learning is rarely used to teach a cognitive skill. Instead, students typically access examples, textbooks and sometimes teachers or peers. However, the instructional material is almost always incomplete. As mentioned earlier, roughly half the pieces of knowledge required to solve problems were not mentioned in the college textbook used in the Chi et al (1989) study. When faced with incomplete instructional material, students need to discover the missing information for themselves. VanLehn (1995c) analyzed the Chi et al (1989) protocols and found learning events in most of the places where information that was missing from the instruction was required for solving problems and explaining examples.

In all these studies, learning events were characterized by long pauses or verbal signs of confusion. (All the studies were based on verbal protocols.) Subjects rarely announced their discoveries in a clear fashion. Even when probed, their explanations were seldom coherent (Siegler & Jenkins, 1989). Nonetheless, their problem solving behaviors changed.

As an illustration of a learning event, consider a physics student quoted in VanLehn (1995c). The student did not know that weight is a kind of force, so he could not figure out how to find the weight of an object even though he had calculated that the force of gravity on the object was 160 pounds. After pausing and complaining a bit, he recalled the textbook equation $F = (W/g)a$ and performed the totally unjustified substitution of g for a, thus deriving the equation $F = W$, which he interpreted as meaning that the object's weight is equal to the force of gravity on it. Amazingly, he had discovered the missing correct principle via a specious derivation. He said, "Right? Is that right? Okay. Um, so I'm going to get 160 pounds. That's the force. Yeah. It kind of makes sense 'cause they, they weigh you in pounds, don't they? That's force." The learning event consisted of reaching an impasse, using purely syntactic algebraic symbol manipulation to hypothesize a new principle (that weight is the force of gravity on an object), then supporting the principle with the fact that pounds is a unit of force, and that Americans express weight in pounds. However, this is not the end of the story. On the next occasion where the subject could use the principle, he failed to do so. This resulted in a units error, which he detected while checking his answer. That reminded him of his new principle (he said, "Oh, wait a minute! Oh, hold the bus. Hold the bus!"), but after a brief pause, he decided the principle didn't apply. However, after he had failed to find another explanation for the units error, he changed his mind and decided the principle did apply. In short, learning a principle in the context of problem solving is much like learning in one of the single-principle experiments reviewed earlier. One may not be reminded of it immediately, but when one deliberately searches memory, the new principle can be retrieved. Yet retrieval of a new principle is not enough. One often has to deliberate about its generality before applying it. Presumably, this deliberation causes generalization.

Learning events can be analyzed in terms of three characteristics:

- What provoked the student to switch attention from the main task to learning? In the illustration, reaching an impasse provoked the physics student to try to derive a new principle. In some situations, most learning events were triggered by impasses (VanLehn, 1991; VanLehn, 1995c). However, when the student already has a correct, operational solution procedure, improvements to the procedure are triggered by some kind of "noticing" process (Siegler & Jenkins, 1989; VanLehn, 1991), which can be modeled (Jones & VanLehn, 1994) but is still not well understood. After solving a problem, some students deliberately reflect on its solution in order to uncover its overall plan or basic idea (VanLehn, 1989; Recker & Pirolli, 1994).

- What kind of reasoning went on during the learning event? In the illustration, the student used both algebraic symbol manipulation and common sense reasoning to derive his new principle. Such "educated guesses" seem often to be based on causal attribution heuristics (Lewis, 1988) or overgeneralizations (VanLehn, 1995c). However, students often use less constrained forms of induction (VanLehn, 1995c), in which case one can observe the prototypicality effects usually found in concept formation experiments (Lewis & Anderson, 1985).

- How easily retrieved is the new principle, and how general is it? Advocates of discovery learning often hope that because students discover principles by themselves, the principles will be easily retrieved and adequately general. As illustrated above, this does not seem to be the case. Principles acquired during learning events are often difficult to retrieve and seem to require deliberate attention before they are general enough (Siegler & Jenkins, 1989; Kuhn & Phelps, 1982; VanLehn, 1989, 1995c).

Progress on understanding learning events has been slow because it is difficult to distinguish them from ordinary problem solving even in the protocols of the most talkative subjects. It has been necessary to have a strong theory of the kind of knowledge that can be learned and a long enough period of observation that one can see slow changes in the usage of individual pieces of knowledge. Nonetheless, the observations to date support the general hypothesis that learning a cognitive skill consists of learning a large number of small pieces of knowledge (including principles, technical details, heuristics, etc.), and that for each one, learning consists of a series of learning events that construct and generalize it.

3.4 Learning from a computer tutor

The discussion of the intermediate stage up to this point has assumed that the learner has no one to talk to and only written materials to refer to. While this constraint simplifies the observation of learning, it makes the instructional situation somewhat atypical of real-world situations where learners can obtain help by raising their hands, picking up the phone or walking down the hall to a colleague's office. As a first step toward understanding cognitive skill acquisition "in the wild," researchers have studied students learning from a very taciturn tutor, namely, a computer.

Although current technology does not allow a computer to converse with students the way human tutors do, computer tutors do have three advantages over instruction based on written materials:

- When working with written materials, students are usually assigned to solve a fix set of problems for each chapter. A computer tutor can select a problem for the student to solve based on the student's performance on preceding problems. Thus, different students will get different problems to solve.

- The computer tutor can point out errors to the student soon after they are committed. Errors are overt, incorrect actions taken by the student. Although tutors can sometimes infer the flaw in the student's knowledge that caused the error, many tutors leave it to the student to find and remedy the sources of their errors.

- The computer tutor can answer certain questions and requests for help, although the dialog is highly constrained.

The impact of each of these three features on learning is discussed in turn.

When the tutor selects problems for the student to solve, a simple policy is to keep giving the student problems until the student has reached mastery. *Mastery* can be defined as answering the most recent N problems with a score higher than M. Some computer tutors (e.g., Anderson et al, 1995) can monitor the usage of individual principles or steps, so they implement a policy that keeps assigning problems until each has been mastered. Varying the thresholds of mastery for individual pieces of knowledge can accurately predict errors (Corbett, Anderson & O'Brien, 1995). Mastery-based tutoring generally causes higher scores on post-tests than tutoring based on fixed problem sets (e.g., Anderson, Conrad & Corbett, 1989), which suggests that different students learn at different rates, which in turn is consistent with their using different studying strategies (section 3.2), different kinds of reasoning during a learning events (section 3.3), and perhaps other factors as well.

Perhaps the most hotly debated issue concerns the control of feedback on errors. The current technology enables students to use the computer as scratch paper. Instead of simply entering their answers to problems, as early computer tutors required, students today can show all their work to the computer. This allows the computer to detect errors soon after they are committed. Many tutors inform the student as soon as an error is detected and prevent them from going on until the error is corrected. Although such feedback makes the tutor easier to build, one wonders what effect it has on learning. Lewis & Anderson (1985) found that immediate feedback that forced students to correct the error before moving on was superior to delayed feedback, but the effect was quite small. Anderson et al (1995) contrasted four feedback policies: (1) no feedback, (2) feedback when the student asked for it, (3) immediate feedback that notified the student as soon as an error was made but did not force the student to correct the error, and (4) immediate feedback that forced immediate correction of the error. In conditions 2 and 3, students were forced to answer the problem correctly before being allowed to continue to the next problem. In condition 4, the feedback also insured that they answered the problem correctly. Only the no-feedback condition, 1, did not force students to correct their mistakes. On post-test measures of learning, students in the no-feedback condition did worse than those in the 3 feedback conditions, and there were no significant differences among the 3 feedback conditions. Thus, it appears that neither the timing of feedback nor its control (student vs. machine) make much difference on whether a student eventually acquires the target knowledge. However, if feedback is completely removed, then students are often unable to correct flaws in their knowledge on their own.

This result is consistent with the studies of learning events, which suggest that constructing or modifying a principle is a deliberate process that is often triggered when students find out they have made a mistake. It should not matter when they learn that they have made a mistake as long as they are able to locate the missing or incorrect knowledge and fix it. However, if they are not forced to correct their errors, then the flaws in their knowledge may remain undiscovered and uncorrected.

Although control of feedback does not seem to affect the basic learning process, there are other aspects of immediate feedback that warrant consideration by the instructional designer. Lewis & Anderson (1985) found that their delayed feedback subjects were much better than the immediate feedback subjects at detecting their own errors. This makes sense, because the immediate feedback subjects had no opportunity to detect their own errors during training. Anderson et al (1995) found that feedback that forced students to correct their errors immediately caused them to complete their training much faster than feedback that lets students choose when to correct their errors. Apparently, students sometimes waste time going down garden paths when they are given the freedom to do so. If instruction was limited to a fixed amount of time, this could hurt their learning.

In the feedback studies just discussed, tutors only notified students of their errors. They did not point out why the students' actions were incorrect nor what should have done instead. This pedagogy is called *minimal feedback*. Several studies have contrasted minimal feedback with feedback designed to be more helpful. Help that is designed to get students to infer the correct action increases their learning rate substantially, compared to minimal feedback (Anderson, Conrad & Corbett, 1989; McKendree, 1990; Mark & Greer, 1995).

On the other hand, adding more help that focuses on what the student did wrong does not appear to offer any advantages (Sleeman et al, 1989). That is, if the feedback is "The right thing to do here is X," then it does little good to add, "You did Y instead. Y is wrong because...." This finding is consistent with studies showing that more errors are caused by missing knowledge rather than incorrect knowledge (VanLehn, 1995c; VanLehn, 1990).

Missing knowledge often causes students to reach an impasse. Rather than seeking help, they often invent an expedient repair, which allows them to continue but risks getting the problem wrong (VanLehn, 1990). Moreover, they often repair the same impasse in different ways on different occasions (VanLehn, 1990). Thus it comes as no surprise that they do not need much convincing in order to get them to abandon the particular repair that led them to receive negative feedback.

Although tutoring does provide a different context in which learning can occur, it appears that the same basic learning processes occur within it as occurred in passive, example-studying instructional situations. However, because learning is based on noticing flaws (both incomplete and incorrect knowledge) and repairing them, tutoring can make learning substantially more efficient by helping students to detect their flaws (via feedback), to rectify the flaws (via help) and to improve the generality and accessibility of the new knowledge (by practicing it until mastery is reached).

4 The final phase: practice effects

The intermediate phase is officially over when students can produce error-free performances. However, this is not the end of their learning. Continued practice causes increases in speed and accuracy. This section reviews these phenomena.

4.1 The power law of practice and other general effects

Perhaps the most ubiquitous finding is the famous power law of practice. The time to do a task goes down in proportion to the number of trials raised to some power. In an influential review, Newell & Rosenbloom (1981) found that the power law applies to simple cognitive skills as well as perceptual-motor skills. Several studies of complex cognitive skills (e.g., Anderson, Conrad & Corbett, 1989; Anderson & Fincham, 1995) found that the speed of applying individual components of knowledge increased according to a power law, thus indicating that practice benefits exactly the pieces of knowledge used and not the skill as a whole. Accuracy also increases according to a power law, at least on some tasks (Logan, 1988; Anderson & Fincham, 1995).

Several theories of the power law have been advanced. Anderson (1993) claims that the speed up is due to two mechanisms: knowledge is converted from a slow format (declarative knowledge) into a fast format (procedural knowledge), and the speed of individual pieces of procedural knowledge also increases with practice. Newell and Rosenbloom (1981; Newell, 1990) claim that small, general pieces of knowledge are gradually composed together (chunked) to form large, specific pieces of knowledge, thus allowing the same task to be accomplished by applying fewer pieces of knowledge. Logan (1988) advances an instance-based theory and reviews several other proposals.

With substantial practice, perceptual-motor skills can be sometimes become *automatic*. Automatic processing is fast, effortless, autonomous and unavailable to conscious awareness (Logan, 1988). One study, using a dual-task paradigm, suggested that different parts of a complex cognitive skill (troubleshooting digital circuits) became automatic after differing amounts of practice (Carlson et al. 1990). However, the skill as a whole never became as automatic as driving one's car despite substantial practice (347 problems) with only 3 circuits.

It is likely that other effects known to occur with perceptual-motor skills also occur with cognitive skills (e.g., massed vs. distributed practice; warm up effects; randomized vs. blocked practice). Proctor & Dutta (1995) review research on many types of skill and find several general effects.

4.2 Replacing mental calculations by memory retrieval

Some cognitive skills are deterministic calculations in that the answer is completely and uniquely determined by the inputs. Mental arithmetic calculations are examples of such deterministic calculations. If subjects are given enough practice with a particular input, then they will eventually just retrieve the output from memory rather than mentally calculate it. This fairly uncontentious phenomenon has been found with a number of simple mental tasks (Logan, 1988; Anderson & Fincham, 1994; Healy et al., 1995) as well as deterministic subprocedures of complex tasks (Carlson et al, 1990). Not only do the subjects report this change in strategy, but they are much faster in responding to practiced inputs than unpracticed ones.

The change in strategy from calculation to retrieval could be taken as an explanation for power-law increases in speed an accuracy. However, Rickards (reported in Healy et al, 1995) had subjects report after each trial whether they had calculated the answer or retrieved it. The calculation trials' latencies fit one power curve well, and the retrieval trails' latencies fit a second curve well, but the fit of a power-law curve to all trials' latencies was poor. Logan (1988) also found evidence for two power curves. Thus, it appears that power law learning mechanisms operate separatedly on both the mental calculation and the retrieval strategies. This hypothesis is also consistent with a study by Carlson & Lundy (1992), who found, using relatively complicated mental calculations, that subjects sped up rapidly when given the same inputs repeatedly, thus promoting use of retrieval. When the inputs varied, thus blocking the retrieval strategy, they still sped up albeit more slowly. The two affects were additive. Thus, it appears that changing from a calculation strategy to a direct retrieval strategy is not in itself a sufficient explanation for the power law of practice.

4.3 Transfer of the benefits of practice

Although "transfer" has many meanings, it is used in this section to mean the savings in learning on one task (the transfer task) due to earlier training on a different task (the training task). Transfer is expressed as a ratio: the time saved in learning the transfer task divided by the time spent learning the training task. Thus, if practicing the training task for 20 hours yields equivalent improvement on the transfer task as 20 hours of practice on the transfer task, then there is 20/20 or 100% transfer. If practicing the training task for 20 hours only saves 5 hours of practice on the transfer task, then the transfer is 5/20 or 25%. If prior practice with the training task makes no difference in how long it takes to learn the transfer task, then there is no transfer. If practice on the training task increases the time to learn the transfer task, then there is negative transfer. Singley & Anderson (1989, chapter 1) discuss several ways to measure this type of transfer.

One major, albeit uncontroversial finding is that the degree of transfer can be predicted by the number of pieces of knowledge shared between the training and transfer tasks (e.g., Bovair, Kieras & Polson, 1990; Singley & Anderson, 1989). As mentioned earlier, practice benefits individual pieces of knowledge and not the skill as a whole. When practice on the training task speeds up certain subskills, they continue to be fast when used in the transfer task.

However, there are sometimes limits to the amount of transfer that one can obtain even when substantial knowledge is shared between tasks. This occurs because practice can change subjects' strategies for solving problems. Suppose that practice causes subjects to change from strategy A to B on the training task, and from strategy A to C on the transfer task. A little training on the training tasks will affect only strategy A, which is shared with the transfer task. Thus, T hours of practice on the training task will save T hours of practice on the transfer task. However, this only occurs when T is less than the amount of practice that will cause subjects to shift from strategy A to B. Let us suppose this shift occurs with around 25 hours of practice. When T exceeds 25, then subsequent practice only affects strategy B, which is not shared with the transfer task. Thus, vast amounts of training only save 25 hours of practice on the transfer task, so the amount of transfer is $25/T$. Thus, when T is less than 25, then transfer is T/T or 100%, but when T is greater than 25, transfer is only $25/T$, which approaches zero as practice on the training task increases. In short, the more practice on the training task, the less transfer.

One strategy change discussed earlier is that practice can cause subjects to use direct memory retrieval instead of mental calculations. As Logan (1988) and Carlson et al (1990) have demonstrated, this is one way that practice can cause a decrease in transfer. Suppose the training task is to master a mental algorithm with one set of inputs, and the transfer task is to master the same algorithm with a different set of inputs. As long as the subject continues to use the algorithm during training, the effects of that training should transfer. However, as the subject starts to use direct memory retrieval instead of executing the algorithm, increasing the time spent in training will not reduce the time to master the transfer task beyond a certain point. Thus, practice decreases the amount of transfer.

The practice-driven decrease in transfer can also be caused by weaning the subjects from the instructional material, which is another kind of strategy change. As discussed earlier, subjects in the intermediate phase of their training often refer to the instructional materials (usually the examples) as they solve problems. However, the frequency of these references declines with practice, and eventually subjects no longer refer to the instructional materials. Suppose that the training and transfer tasks share some of their instructional materials (e.g., the examples). Even a little practice on the training task familiarizes subjects with the shared instructional materials, and thus saves them time in learning the transfer task. However, with increasing practice on the training task, subjects gain no further benefit on the transfer task. Thus, more practice causes less transfer. This effect has been found in a variety of cognitive skills (Singley & Anderson, 1989; Anderson & Fincham, 1995). Anderson and his colleagues call this phenomena the *use specificity* of transfer and consider it prime evidence for the distinction between declarative and procedural knowledge.

Suppose the training task is actually a part of the transfer task, for instance, as mental addition is a part of mental multiplication. Intuitively, it seems that transfer should be high regardless of whether the training task is practiced for minutes or days. However, this does not seem to be the case. For relatively small amounts of practice, the amount of overlap between tasks accurately predicts the amount of transfer (Bovair, Kieras & Polson, 1990; Singley & Anderson, 1989). However, when the training task is practiced for hundreds of trials, only a few trials of practice on the transfer task are saved, even though the training task is arguably a part of the transfer task (Frensch & Geary, 1993). Needless to say, more research is required on this important point, because whole educational systems are founded on the premise that training on basic skills facilitates learning practical skills that employ the basic skills as subprocedures. Frensch & Geary's result suggests that training basic skills past a certain point is wasteful.

In short, a great deal of practice on one task (the training task) will help subjects perform that task, but it usually only saves them moderate amounts of time in learning a second task (the transfer task). In fact, the head start that they get on learning the second task after a great deal of practice on the first task is about the same as the head start they would get if they only practiced the first task a moderate amount. The degree of transfer observed, and how it varies with practice on the first task, depends on exactly what is shared between the tasks and on when practice causes changes in problem solving strategies.

4.4 Negative transfer

Negative transfer occurs when the training task interferes with learning the transfer task and slows the learning down. For instance, mastering one text editor often seems to interfere with learning a similar text editor that has different commands.

The amount of transfer seems to be inversely proportional to the amount of practice on the transfer task. Using text editors designed to interfere with each other's learning, Singley & Anderson (1989) showed that negative transfer occurs mostly during the early stage of learning the transfer task. If the learner learns the transfer task by receiving immediate feedback whenever they do something wrong (including habits carried over from the training task), then they rapidly acquire the correct responses. On the other hand, uncorrected responses (e.g., ones that are merely inefficient and not incorrect) persist and can cause negative transfer even in the later stages of learning the transfer task.

As discussed in the preceding section, it is often the case that one gets the same time savings in learning the transfer task regardless of the amount of practice on the training task beyond a certain minimum. It would be interesting to see if this were true of negative transfer as well. That is, would moderate amounts of training delay the learning of the transfer task as much as vast amounts of training? To put it colloquially, is negative transfer due to bad ideas, bad habits or both?

5 The frontiers of cognitive skill acquisition research

Historically, cognitive skill acquisition research started with simple knowledge-lean puzzle tasks, moved on to problems (such as the one of Table 1) that can be solved with a single principle from a knowledge-rich task domain, and most recently has focused on chapter-sized slices of a knowledge-rich task domain. The next step in this progression is to study the acquisition of even larger pieces of knowledge, such as a whole semester's worth of physics or programming. Some early work along these lines has been done where students learned from a tutoring system (e.g., Anderson et al., 1995), but these studies have examined acquisition of only the procedural aspects of the skill. The next step would be to examine how concepts, mental models and factual knowledge evolve together with the more procedural aspects.

As done in this review, research on cognitive processing during the intermediate phase of skill acquisition can be conveniently divided into research on the acquisition of a single principle, and research on the acquisition of collections of principles and other knowledge. The single-principle research seems to have answered many of the initial questions concerning retrieval, mapping and generalization. The remaining areas of uncertainty lie with application. It is likely that the students use the same diverse set of methods for applying principles as they use for drawing analogies to complex written examples (VanLehn, 1995a).

Learning from examples has been the focus of research on the intermediate phase of acquisition for complex, chapter-sized pieces of knowledge. However, most studies have focused only on passive forms of instruction, where the student studies written material with no help from a teacher, peer or tutor. The next step would be to find out how students learn from more interactive forms of instruction. For instance, when students receive feedback, it seems plausible that poor learners merely change their answer without thinking about why they got it wrong, whereas good learners would try to find and repair the flaw in their knowledge that caused the error.

The final phase of skill acquisition has become a battleground for some of the major theoretical questions of the day. The controversy between instance-based (e.g., Logan, 1988), procedural-declarative (Anderson, 1993) and other theories of memory have been mentioned already. The debate over implicit learning (e.g., Berry & Broadbent, 1984) also bears most strongly on the final phase.

The first responsibility of research on cognitive skill acquisition ought to be to account for the differences between experts and novices. Surprisingly, it has not done so yet. Ericsson & Lehmann (this volume) argue that in many task domains, experts can mentally plan solutions to problems that novices can only solve concretely. For instance, Koedinger & Anderson (1990) found that expert geometers plan solutions to geometry proof problems using abstract, diagramatic schemas. Even though a proof would consist of dozens of lines, the plan might have only one or two schema applications. The experts are able to do such planning in working memory. Novices, on the other hand, build proofs a line at a time, writing down conclusions as they go. The ability to rapidly plan solutions seems to develop rather quickly (Carlson et al, 1990).

As Koedinger & Anderson point out, no existing model of skill acquisition, including Anderson's ACT*, can account for the acquisition of domain-specific planning skill. Existing models have concentrated on explaining the power law of practice and the increasing specificity of transfer. These results call for mechanisms that convert knowledge into ever more specific forms. In contrast, the development of planning skill calls for converting knowledge into more abstract forms. Artificial Intelligence has produced may computationally sufficient mechanisms for abstraction. What our science needs is empirical work that discriminates among the many possible ways that planning could develop. For instance, are novices held back by a lack of knowledge of planning schemas, or do they know the schemas but cannot apply them mentally due to working memory limitations?

6 Acknowledgements

The preparation of this chapter was supported by the Cognitive Sciences Division of the Office of Naval Research under grant N00014-92-J-1945. I gratefully acknowledge the comments of Micki Chi and Patricia Albacete.

References

Anderson, J. (1983). *The Architecture of Cognition.* Harvard University Press, Cambridge, MA.

Anderson, J. R. (1993). *Rules of the Mind.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Anderson, J. R., Conrad, F. G., and Corbett, A. T. (1989). Skill acquisition and the LISP tutor. *Cognitive Science*, 14(4):467–505.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., and Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4(2):167–207.

Anderson, J. R. and Fincham, J. M. (1994). Acquisition of procedural skills from examples. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 20(6):1322–1340.

Bernardo, A. B. I. (1994). Problem-specific information and the development of problem-type schemata. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 20(2):379–395.

Berry, E. C. and Broadbent, D. E. (1984). On the relationship between task performance and associated verbalizable knowledge. *The Quarterly Journal of Experimental Psychology*, 36A:209–231.

Bielaczyc, K., Pirolli, P., and Brown, A. L. (1994, April). Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem-solving. Technical report, University of California at Berkeley. Report CSM-7. In press, Cognition and Instruction.

Bielaczyc, K. and Recker, M. (1991). Learning to learn: The implications of strategy instruction in computer programming. In Birnbaum, L., editor, *The International Conference on the Learning Sciences*, pages 39–44. Association for the Advancement for Computing in Education, Charlottesville, VA.

Bovair, S., Kieras, D. E., and Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cogitive complexity analysis. *Human-Computer Interaction*, 5:1–48.

Bransford, J. D., Franks, J. J., Vye, N. J., and Sherwood, R. D. (1989). New approaches to instruction: Because wisdom can't be told. In Vosniadou, S. and Ortony, A., editors, *Similarity and analogical reasoning*, pages 470–497. Cambridge University Press, New York.

Brown, A. and Kane, M. (1988). Preschool children can learn to transfer: Learning to learn and learning from example. *Cognitive Psychology*, 20:493–523.

Carlson, R. A., Khoo, B. H., Yaure, R. G., and Schneider, W. (1990). Acquisition of a problem-solving skill: Levels of organization and use of working memory. *Journal of Experimental Psychology: General*, 110(2):193–214.

Carlson, R. A. and Lundy, D. (1992). Consistency and restructuring in learning cognitive procedural sequences. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 19(1):127–141.

Carroll, W. M. (1994). Using worked examples as an instructional support in the algebra classroom. *Journal of Educational Psychology*, 86(3):360–367.

Catrambone, R. (1994a). The effects of labels in examples on problem solving transfer. In Ram, A. and Eiselt, K., editors, *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, pages 159–164. Lawrence Erlbaum Associates, Hillsdale, NJ.

Catrambone, R. (1994b). Improving examples to improve transfer to novel problems. *Memory and Cognition*, 22(5):606–615.

Catrambone, R. and Holyoak, K. (1989). Overcoming contextual limitations on problem-solving transfer. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 13(6):1147–1156.

Catrambone, R. and Holyoak, K. (1990). Learning subgoals and methods for solving probability problems. *Memory and Cognition*, 18(5):593–603.

Catrambone, R. and Holyoak, K. J. (1987). Transfer in problem solving as a function of the procedural variety of training examples. In *Proceedings of the 9th Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Chi, M., Bassok, M., Lewis, M., Reimann, P., and Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 15:145–182.

Chi, M., de Leeuw, N., Chiu, M.-H., and LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, 18:439–477.

Chi, M. and VanLehn, K. (1991). The content of physics self-explanations. *The Journal of the Learning Sciences*, 1:69–105.

Cooper, G. and Sweller, J. (1987). Effects of schema acquisition and rule automation on mathematical problem-solving transfer. *Journal of Educational Psychology*, 79(4):347–362.

Corbett, A. T., Anderson, J. R., and O'Brien, A. T. (1995). Student modeling in the ACT programming tutor. In Nichols, P. D., Chipman, S. F., and Brennan, R. L., editors, *Cognitively Diagnostic Assessment*, pages 19–42. Lawrence Erlbaum Associates, Hillsdale, NJ.

Cummins, D. D. (1992). Role of analogical reasoning in the induction of problem categories. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 18(5):1103–1124.

Duncan, C. P. (1959). Recent research on human problem solving. *Psychological Bulletin*, 56(6):397–429.

Ericsson, K. A. and Lehmann, A. C. (1996). Expert and exceptional performance: Evidence of maximal adaptation to task constraints. This volume.

Faries, J. and Reiser, B. (1988). Access and use of previous solutions in a problem solving situation. In Patel, V. and Groen, G., editors, *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, pages 433–439. Lawrence Erlbaum Associates, Hillsdale, NJ.

Ferguson-Hessler, M. and de Jong, T. (1990). Studying physics texts: Differences in study processes between good and poor solvers. *Cognition and Instruction*, 7:41–54.

Frensch, P. and Geary, D. C. (1993). Effects of practice on component processes in complex mental addition. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 19(2):433–456.

Gick, M. and Holyoak, K. (1980). Analogical problem solving. *Cognitive Psychology*, 12:306–355.

Gick, M. and Holyoak, K. (1983). Schema induction and analogical transfer. *Cognitive Psychology*, 15:1–38.

Glaser, R. and Bassok, M. (1989). Learning theory and the study of instruction. *Annual Review of Psychology*, 40:631–636.

Healy, A. F. and L. E. Bourne, J. (1995). *Learning and Memory of Knowledge and Skills: Durability and Specificity.* Sage Publications, Thousand Oaks, CA.

Holyoak, K. J. and Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory and Cognition*, 15(4):332–340.

Jones, R. M. and VanLehn, K. (1994). Acquisition of children's addition strategies: A model of impasse-free, knowledge-level learning. *Machine Learning*, 16(1/2):11–36.

Kanhey, H. (1993). *Problem Solving: Current issues (Second edition).* Open University Press, Buckingham.

Karmiloff-Smith, A. and Inhelder, B. (1975). If you want to get ahead, get a theory. *Cognition*, 3(1):195–212.

Koedinger, K. and Anderson, J. R. (1990). Abstract planning and perceptual chunks: Elements of expertise in geometry. *Cognitive Science*, 14:511–550.

Kuhn, D. and Phelps, E. (1982). The development of problem-solving strategies. *Advances in Child Development and Behavior*, 17:1–44.

LeFevre, J. and Dixon, P. (1986). Do written instructions need examples? *Cognition and Instruction*, 3:1–30.

Lewis, C. (1988). Why and how to learn why: Analysis-based generalization of procedures. *Cognitive Science*, 12(2):211–256.

Lewis, M. W. and Anderson, J. R. (1985). Discrimination of operator schemata in problem solving: Learning from examples. *Cognitive Psychology*, 17:26–65.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, 95(4):492–527.

Lovett, M. C. (1992). Learning by problem solving versus by examples: The benefits of generating and receiving information. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Mark, M. A. and Greer, J. E. (1995). The vcr tutor: Effective instruction for device operation. *The Journal of the Learning Sciences*, 4(2):209–246.

McKendree, J. (1990). Effective feedback content for tutoring complex skills. *Human-Computer Interaction*, 5:381–413.

Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA.

Newell, A. and Rosenbloom, P. (1981). Mechanisms of skill acquisition and the law of practice. In Anderson, J., editor, *Cognitive Skills and Their Acquisition*, pages 1–56. Lawrence Erlbaum Associates, Hillsdale, NJ.

Newell, A. and Simon, H. A., editors (1972). *Human Problem Solving*. Prentrice-Hall, Inc., Englewood Cliffs, NJ.

Norman, D. A. (1981). Categorization of action slips. *Psychological Review*, 88(1):1–15.

Novick, L. (1988). Analogical transfer, problem similarity and expertise. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 14:510–520.

Novick, L. and Holyoak, K. (1991). Mathematical problem solving by analogy. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 17(3):398–415.

Pirolli, P. (1991). Effects of examples and their explanations in a lesson on recursion: A production system analysis. *Cognition and Instruction*, 8(3):207–260.

Pirolli, P. and Anderson, J. (1985). The role of learning from examples in the acquisition of recursive programming skills. *Canadian Journal of Psychology*, 39:240–272.

Pirolli, P. and Bielaczyc, K. (1989). Empirical analyses of self-explanation and transfer in learning to program. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, pages 459–457, Hillsdale, NJ. Lawrence Erlbaum Associates.

Pirolli, P. and Recker, M. (1994). Learning strategies and transfer in the domain of programming. *Cognition and Instruction*, 12(3):235–275.

Pressley, M., Wood, E., Woloshyn, V., Martin, V., King, A., and Menke, D. (1992). Encouraging mindful use of prior knowledge: Attempting to construct explanatory answers facilitates learning. *Educational Psychologist*, 27:91–109.

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

Recker, M. M. and Pirolli, P. (1995). Modeling individual differences in students' learning strategies. *The Journal of the Learning Sciences*, 4(1):1–38.

Reed, S. K., Dempster, A., and Ettinger, M. (1985). Usefulness of analogous solutions for solving algebra word problems. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 11:106–125.

Reed, S. K., Willis, D., and Guarino, J. (1994). Selecting examples for solving word problems. *Journal of Educational Psychology*, 86(3):380–388.

Ross, B. (1984). Remindings and their effects in learning a cognitive skill. *Cognitive Psychology*, 16:371–416.

Ross, B. (1987). This is like that: The use of earlier problems and the separation of similarity effects. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 13:629–639.

Ross, B. (1989). Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 15(3):456–468.

Ross, B. and Kennedy, P. (1990). Generalizing from the use of earlier examples in problem solving. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 16(1):42–55.

Siegler, R. S. and Jenkins, E. (1989). *How Children Discover New Strategies.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Singley, M. and Anderson, J. (1989). *The transfer of cognitive skill.* Harvard University Press, Cambridge, MA.

Sleeman, D., Kelley, A. E., Martinak, R., Ward, R. D., and Moore, J. L. (1989). Studies of diagnosis and remediation with high school algebra students. *Cognitive Science*, 13:551–568.

Sweller, J. and Cooper, G. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, 2:59–89.

VanLehn, K. (1989). Problem solving and cognitive skill acquisition. In Posner, M., editor, *Foundations of Cognitive Science*, pages 526–579. MIT Press, Cambridge, MA.

VanLehn, K. (1991). Rule acquisition events in the discovery of problem solving strategies. *Cognitive Science*, 15(1):1–47.

VanLehn, K. (1995a). Analogy events: How examples are used during problem solving. Submitted for publication.

VanLehn, K. (1995b). Looking in the book: The effects of example-exercise analogy on learning. Submitted for publication.

VanLehn, K. (1995c). Rule learning events in the acquisition of a complex skill. Submitted for publication.

VanLehn, K. and Jones, R. (1993). Learning by explaining examples to oneself: A computational model. In Chipman, S. and Meyrowitz, A., editors, *Cognitive Models of Complex Learning*, pages 25–82. Kluwer Academic Publishers, Boston, MA.

VanLehn, K. and Jones, R. M. (1995). Is the self-explanation effect caused by learning rules, schemas or examples? Submitted for publication.

VanLehn, K., Jones, R. M., and Chi, M. T. H. (1992). A model of the self-explanation effect. *The Journal of the Learning Sciences*, 2(1):1–59.

Voss, J., Wiley, J., and Carretero, M. (1995). Acquiring intellectual skills. *Annual Review of Psychology*, 46:155–181.

W.-K. Ahn, W. F. B. and Mooney, R. J. (1992). Schema acquisition from a single example. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 18(2):391–412.

Zhu, X. and Simon, H. A. (1987). Learning mathematics from examples and by doing. *Cognition and Instruction*, 4(3):137–166.