Kirsh, David. "Problem solving and situated cognition." In Robbins, P., & Aydede, M. (Eds.). (2009). The Cambridge handbook of situated cognition (pp. 264-306). Cambridge: Cambridge University Press.

CHAPTER 15

Problem Solving and Situated Cognition

David Kirsh

Introduction

In the course of daily life we solve problems often enough that there is a special term to characterize the activity and the right to expect a scientific theory to explain its dynamics. The classical view in psychology is that to solve a problem a subject must frame it by creating an internal representation of the problem's structure, usually called a problem space. This space is an internally generable representation that is mathematically identical to a graph structure with nodes and links. The nodes can be annotated with useful information, and the whole representation can be distributed over internal and external structures such as symbolic notations on paper or diagrams. If the representation is distributed across internal and external structures the subject must be able to keep track of activity in the distributed structure. Problem solving proceeds as the subject works from an initial state in this mentally supported space, actively constructing possible solution paths, evaluating them and heuristically choosing the best. Control of this exploratory process is not well understood, as it is not always systematic, but various heuristic search algorithms have been proposed and some experimental support has been provided for them.

Situated cognition, by contrast, does not have a theory of problem solving to compete with the classical view. It offers no computational, neuropsychological, or mathematical account of the internal processes underlying problem cognition. Nor does it explain the nature of the control of external processes related to problem solving. Partly this is a matter of definition. Problems are not regarded to be a distinct category for empirical and computational analysis because what counts as a problem varies from activity to activity. Problems do arise all the time, no matter what we are doing. But from a situated cognition perspective these problems should not be understood as abstractions with a formal structure that may be the same across different activities. Each problem is tied to a concrete setting and is resolved by reasoning in situation-specific ways, making use of the material and cultural resources locally available. What is called a problem, therefore, depends on the discourse of that activity, and so in a sense, is socially constructed. There is no natural kind called "problem" and no natural kind process called "problem solving" for psychologists to study. Problem solving is merely a form of reasoning that, like all reasoning, is deeply bound up with the activities and context in which it takes place. Accordingly, the situational approach highlights those aspects of problem solving that reveal how much the machinery of inference, computation, and representation is embedded in the social, cultural, and material aspects of situations.

This critical approach to problem solving is what I shall present first. In Part 1 I discuss the assumptions behind the classical psychological theory. In Part 2 I present the major objections raised by those believing that cognition must be understood in an embodied, interactive, and situated way, and not primarily as a cognitive process of searching through mental or abstract representations. There is a tendency in the situated cognition literature to be dismissive of the classical view without first acknowledging its flexibility and sophistication. Accordingly, I present the classical account in its best form in an effort to appreciate what parts may be useful in a more situated theory. In Part 3 I collect pieces from both accounts, situated and classical, to move on to sketch a more positive theory - or at least provide desiderata for such a theory - though only fragments of such a view can be presented here.

PART 1: THE CLASSICAL THEORY

1. Newell and Simon's Theory

In an extensive collection of papers and books, Herbert Simon, often with Allen Newell, presented a clear statement of the now-classical approach to problem solving (see, among others, Newell & Simon, 1972). Mindful that science regularly proceeds from idealization, Simon and Newell worked from the assumption that a theory based on how people solve well-defined problems can be stretched or augmented to explain how people solve problems that are ill defined, which they recognized a large class of problems to be.

To develop their theory they presented subjects with a collection of games and puzzles with unique solutions or solution sets. Having a correct answer - a solution set - is the hallmark of a problem being well defined. Problems were posed in contexts in which the experimenter could be sure subjects had a clear understanding of what they had to solve. Games and puzzles were chosen because they are self-contained; it is assumed that no special knowledge outside of what is provided is needed to solve them. These sorts of problems have a strict definition of allowable actions (you move your pawn like this), the states these actions cause (the board enters this configuration), and a strict definition of when the game or puzzle has been solved, won, or successfully completed (opponent's king is captured). It was assumed that subjects who read the problem would be able to understand these elements and create their internal representation. Such problems are both well defined and knowledge lean, as "everything that the subject needs to know to perform the task is presented in the instructions" (VanLehn, 1989, p. 528). No special training or background knowledge is required.

2. Task Environment

In the classical theory, the terms ***problem*** and ***task*** are interchangeable. Newell and Simon introduced the expression ***task environment*** to designate an abstract structure that corresponds to a problem. It is called an environment because subjects who improve task performance are assumed to be adapting their behavior to some sort of environmental constraints, the fundamental structure of the problem. It is abstract because the same task environment can be instantiated in very different ways. In chess, for example, the task environment is the same whether the pieces are made of wood or silver or are displayed on a computer screen. Any differences arising because agents need to interact differently in different physical contexts are irrelevant. It does not matter whether an agent moves pieces by hand, by mouse movements, by requesting someone else to make the move for them, or by writing down symbols and sending a description of their move by mail. Issues associated with solving these movement or communication subtasks belong to a different problem.

A task environment, accordingly, delineates the core task. It specifies an underlying structure that determines the relevant effects of every relevant action that a given agent can perform. This has the effect that if two agents have different capacities for action they face different task environments. When four-legged creatures confront an obstacle, they face a different locomotion problem than two-legged creatures, and both problems are different from the locomotion problem the obstruction poses to a snake. Thus, two agents operating in the same physical environment, each facing the same objective - get from a to b - may face different task environments because of their different capacities. Their optimal path may be different. Moreover, of all the actions a creature or subject can perform, the only ones that count as task relevant are the ones that can, in principle, bring it closer to or farther from an environmental state meeting the goal condition. It is assumed that differences in expertise and intellectual ability affect search and reasoning rather than the definition of the task itself.

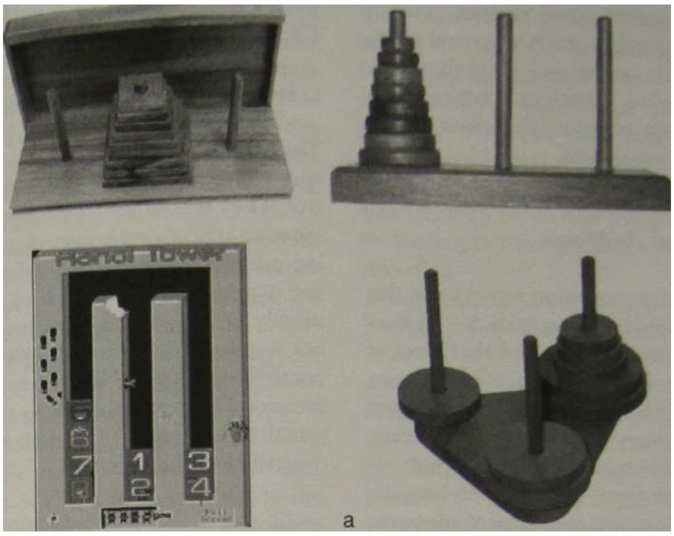
Task environments are theoretical projections that let researchers interpret problem-solving activity in concrete situations. They identify what counts as a move in a problem (for a given agent). As such, they impose a powerful filter over the way a researcher interprets subjects' actions. Scratching one's head during chess, for instance, is an action that would be interpreted by a researcher as irrelevant to the game. It not only would lie outside the task environment of chess construed as the set of possible chess moves but also would be treated as having no relevance to the game in any other way - an epiphenomenon. The same would apply to other things nonexperts do when they play, such as putting a finger on a piece, trying if possible actions on the board, using pencil and paper, talking to oneself, or consulting a book (if allowed at all). All are assumed irrelevant to task performance. They may occur while a subject is working on a problem, or while playing chess, but, according to the classical account, they are not literally part of problem-solving activity. This is obviously a point of dispute for situationalists, as many of these actions are regularly observed during play, and they may critically affect the success of an agent.

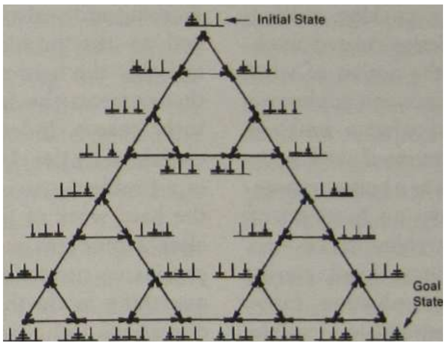
3. Problem Space

Task environments are differentiated from problem spaces, the representation subjects are assumed to mentally construct when they understand a task correctly. This problem-space representation might be distributed over external resources. It encodes the following:

- The current state of the problem. At the beginning this is the initial state.
- A representation of the goal state or condition - though this might be a procedure or test for recognizing when the goal has been reached, rather than a declarative statement of the goal.
- Constraints determining allowable moves and states, hence the nodes and allowable links of the space I these too may be specified implicitly in procedures for generating all and only legal moves rather than explicitly in declarative statements.
- Optionally, other representations that may prove useful in understanding problem states or calculating the effects of action.

Some of these other representations encode knowledge of problem-solving methods, heuristics, or metrics specific to the current task environment. Others encode





b

Figure **15.1.** The different versions of the Tower of Hanoi shown in 15.1a share the same abstract task environment, shown in the graph structure displayed in 15.1b. All the versions have the same legal moves in an abstract sense, the goal of the game is the same, and the strategies for completion are the same. At a more microscopic level, moving heavy pieces in one game may require additional planning, but these extra moves and extra plans are not thought to be part of the game. Because the game is defined abstractly, any differences in the action repertoire of an agent are irrelevant. In other tasks we base our analysis of the task on the actions the agent can perform, so that more powerful agents may face different tasks than less powerful ones. Choice of level of abstraction is a theoretical decision.

methods, heuristics, and metrics that are domain independent, such as general methods of search, measures of when one is getting closer to a goal, and typical ways of overcoming impasses that arise in the solution-finding process.

4. Ill-Defined Problems

Puzzle and game cognition seems to fit this formal, knowledge-lean approach - at least in part. But Simon recognized that most of the problems we encounter in life are not well defined in this formal sense. Some have no unambiguously right answer, the result of applying an operational goal condition to possible solutions. This may be because there are many grades and forms of adequate answer, as is typical of problems arising in architecture, engineering, cooking, writing, and other creative or design-related work. Or it may be because the notion of what constitutes an adequate answer is not known in advance, and part of what a problem solver must learn in the course of working on a problem is what counts as a better answer. Still other problems have no fixed set of operators relevant to a problem space - no fixed set of choice points, fixed consequence function, fixed evaluation function, or well-defined constraints on feasible actions. Think of the problem a painter faces when confronting a blank canvas in a studio with all the paints, media, brushes, and tools he or she might ever want. Goals, operators, choice points, consequence, and evaluation functions are either undetermined by the very nature of the problem, or they have to be learned microgenetically, in the course of activity. The problem is largely being made up as it is being worked on (cf. Reitman, 1964}.

Simon regarded the prevalence of ill-defined problems as a challenge to the classical theory but not an insurmountable one. Cognitive theories should start first with the clear, central cases of problems - which for Simon are well defined and knowledge lean - and then move outward to harder cases.

PART 2: CRITICISM OF THE CLASSICAL THEORY

1. Initial Summary of Objections

The ideas of task environment and problem space have a formal elegance that is seductive. They encourage treating problem solving as an area of psychology that can be studied using existing methods of mathematics and experimentation. But they can also justifiably be attacked from many sides, and not just because efforts to extend the theory to ill-defined problems have been mostly unsuccessful. Four objections that are congenial to a situated approach to cognition deserve close examination.

1.1. Framing and Registration

Framing and registration processes are integral to the problem-solving process and arguably the hardest part of it. The formal theory treats the heart of problem solving to be search. Indeed, this is the only part explained by the classical theory. But search in a problem space only makes sense after the hard work of framing has been done - after a problem has been well posed and put into a searchable graph structure. It is one thing to do this for games where the operations and objectives are typically told to us explicitly. It is another to do this for everyday problems, where we have to decide what is relevant and what is irrelevant. In artificial intelligence, a closely related problem of bounding the scope of what needs to be considered in planning, reasoning, and solving problems is called the "frame problem," and it remains an open question how people do this.

Moreover, what is the justification for treating the abstraction or framing part of problem solving to be separate and unconnected from the problem-solving part, which is assumed to be search? It may seem intuitive to see problem solving as having parts: recognize a problem in a concrete situation; abstract, frame, or bound the problem; find a solution; and reinterpret the solution in the concrete setting. It may seem intuitive that we can modularize these parts and study each component. But whether justified or not - and there are good reasons to challenge the modularization of steps - why accept that the locus of difficulty, the real challenge of problem solving, concerns the search part? Framing is notoriously hard, and so is registration.

To understand the registration problem, imagine yourself in a shopping mall, standing in front of a wall map, trying to find a path from your current location to a specific store. Which is harder: figuring out where you are relative to the map, assuming the map does not have an icon with a "You are here" label, or finding a path from *a* to *b* on the map? For most of us finding the path is the easy part. That is the part that is analogous to search in a problem space. It is far harder to figure out where you are and then translate the path you found back into action in the world. Those are the registration parts: connecting the abstract search space (whether internal or external) to the real world, and then reinterpreting the results of search, or some other action performed on an abstract representation, back into domain-specific terms.

Given the interactive nature of problem solving, the back-and-forth process of acting, observing the result, and then thinking of the next move, agents almost never do all their work in a problem space and then act in the world. They constantly translate moves in their abstract problem space into actions in their concrete context and back again. How subjects frame and interpret a problem therefore is essential to how they will proceed and how easy it is to translate between problem space and world. The more abstract a problem space, the more distant it is from the specifics of the current situation, and the harder this translation process is. Think of the distance between a recipe in a cookbook and its concrete execution by a cook in the kitchen. The recipe represents a solution to the problem of creating a certain dish given certain ingredients. But when cooks execute a recipe they go back and forth between the paper representation and their kitchen. Why can't they just remember the steps and proceed without consulting and reconsulting the recipe? Plan and execution are connected in nonsimple ways. The interim effects of following a recipe alert a cook to details of the steps that need close attention. This interactive process of going back and forth, between world and representation (recipe), shows that there are two sides to the registration problem: encoding and decoding.

Registration and framing are related because in registering a problem one also has to find a way of tying concrete elements of a situation with a problem representation. Framing adds a further element: a bias on the knowledge that is relevant. When people think about something they see as problematic, they typically frame their difficulty in terms of their immediate understanding of their situation, an understanding that comes with preconceptions of what is relevant and potentially useful. This is often constraining. Problems of cooking, for instance, are framed in terms of ingredients, flame size, and pots and pans, rather than in terms of concepts in chemistry (e.g., reaction potential, catalyst) we may have learned in school and that are, in principle, relevant to understanding the cooking process. Expert chemists may bring such domain-external views to the cooking process. And expert mathematicians or expert modelers may bring the capacity to neatly formalize the concrete. But for the rest of us it is hard to get beyond the concrete to the abstract and general. If we could appreciate the abstract in the concrete, we would recognize analogies and be able to transfer learning from one domain to another more readily than we do. The reason we do not is because our understanding of problems is usually tied to the resources and tools at hand. We are hampered by the mindset appropriate to the setting in which our activity takes place.

Given the way problems arise in natural contexts, the burden of explanation ought to lie in psychology to show both that (a) people do have an abstract problem space representation of problems they solve, and (b) the hard part of problem solving is not to be found in the process of going back and forth between situational understanding and problem space understanding, but in search. This is the challenge which greater attention to the processes of framing and registration pose to the classical view. To my mind it has never been answered.

1.2. *Interactivity and Epistemic Activity*

Examination of actual problem solving in ecologically natural contexts as opposed to white-room environments reveals a host of interactions with resources and cultural elements that figure in the many phases of problem solving, such as understanding the problem, exploring its scope and constraints, getting a sense of options, and developing a metric for evaluating progress toward a solution. People generate a range of intermediate structures. In reducing problem solving to search in a problem space, the classical approach minimizes and misunderstands the complexity and centrality of local interaction.

There is much more going on during problem solving than searching in an abstracted problem space. Most of these actions-interactions lie outside the narrow definition of the problem. Although this echoes the first objection in stressing that problem solving is not reducible to search, it pushes that argument further by focusing on the nature of agent-environment interaction during problem solving. People do many more task-relevant things when problem solving than those allowed for in the strict definition of their task or problem. The notion of a task environment is far too narrow. These task-exogenous actions affect both the process and success of problem solving. Addressing this issue requires ethnographic attention to the real-world details of problem solving.

1.3. *Resources and Scaffolds*

Once focus shifts from puzzles to real problems arising in everyday environments, it is apparent that subjects have access to cultural products - tools, measuring devices, graph paper, calculators, algorithms, tricks of the trade, free advice - that make their reasoning job easier. Even when no problem aids are lying around, the type of problem encountered is not a worst-case problem but a simpler version of a problem that only in its general form is hard to solve. It is well known that problems that are computationally complex when conceived in their general form invariably have many special forms that are quite easy to solve. Usually these are the ones people actually confront, and posing a problem in its more general form, as so often is done in the classical approach makes the problem harder, encouraging cognitive scientists to propose solution methods that people do not have to follow.

It is not an accident that we encounter special cases. We live most of our life in constructed environments. Layers of artifacts saturate almost every place we go, and there are preexisting practices for doing things. These artifacts and practices have been designed, or have coevolved, to make us smarter, to make it easier for us to solve our problems and perform our habitual tasks. Everywhere there are scaffolds and other resources to simplify problem solving, including people to ask. Part of what we learn is how to use these resources and participate in the relevant practices. Approaching a problem as if it must be posed in its general form ignores the efficiencies and kludges that typify natural beings living in worlds scaffolded and designed for them. It supposes that our main problem-solving skills are tied to search, when in fact they may be more closely related to our ability to manage our artifacts, make effective use of scaffolds, and conform to practice. To focus on the 3 percent of problems only some of us solve risks misunderstanding the remaining 97 percent of problems we all solve.

1.4. *Knowledge Rich*

Most problems people face in daily life are not like knowledge-lean problems in which all relevant aspects of each problem can be given in a compact problem statement.

Naturally occurring problems rarely occur in a vacuum, where all an agent needs to bow can be encapsulated in a few simple sentences. We typically bring more knowledge and expertise than formal accounts of problem solving discuss. Consider cooking, cleaning, shopping, gardening, the tasks confronted in offices that involve computer applications, or editing documents. In each case an intelligent novice performs less well than experienced participants. It might be that experience can be reduced to familiarity with search heuristics, domain metrics, and the like. But much surely has to do with knowing how to pose, view, dissolve, and work around problems, and knowing what is most effective in specific situations and how to coordinate the use of local resources - a deep knowledge of cases. Theories of knowledge-rich problem solving have become important in the literature since the 1980s. But even these studies place too little emphasis on the centrality of resources, scaffolds, interactivity, and cultural support. Almost none explain the process by which people understand problems.

In the next sections I will develop each argument further, calling attention to supporting articles in both the situated and classical literature where many of these concerns have been recognized but left unanswered.

2, Framing and Registration

2.1. *Framing*

The heart of the framing and encoding argument is that natural problems arise in concrete settings where agents are already operating in activity-specific provinces of meaning. It matters whether an agent is playing chess by mail or playing chess with a young child using Disney characters. The context affects the way the game is conceptualized and framed (e.g., chess for competition, chess for teaching beginners). This framing colors choice, evaluation, and local objectives, all factors involved in creating a problem space. In tasks that are less abstract than chess the setting and local resources matter even more. They activate an interpretive framework that primes agents to look for and conceptualize features of their environment in activity-specific ways, biasing what they see as problematic and what they see as the natural or at-hand resources available to solve such problems. Problems, goals, operators, and representations are not abstract. They arise in concrete settings where agents have certain activities they have to perform. Features of these activity spaces affect the way the problem is represented and framed.

Lave (1988) and others (Rogoff & Lave, 1984) have explored the effects of context on problem conceptualization. In commenting on her well-known ethnography of mathematical activity in supermarkets, Lave wrote: "I have tried... to understand how mathematical activity in grocery stores involved being 'in' the 'store,' walking up and down 'aisles,' looking at 'shelves' full of cans, bottles, packages and jars of food, and other commodities" (Lave, 1996, p. 4). Each of these domain-specific terms has an impact on the way problems are conceptualized and posed.

To show that mathematical activity is not the same across settings, Lave looked at the techniques and methods shoppers in supermarkets use to solve some of their typical problems of choice, such as whether can A is a better buy than can B. She found that even though unit prices are printed on supermarket labels, shoppers rarely check them to decide what to buy. Instead they use less general strategies such as, "Product A would cost $10 for 10 oz., and product B is $9 for 10 oz., hence product B is the cheaper buy." Or faced with a choice between a 5 oz. packet costing $3.29 and a 6 oz. packet priced at $3.59, the shopper would argue, "If I take the larger packet, it will cost me 30 cents for an extra ounce. Is it worth it?"

Why do shoppers ignore unit price? It is clear that they are not indifferent to unit price because they usually use strategies that involve price comparison between specific items. But as retailers well know, the actual problem a shopper solves has many more variables. When people make a decision about what to take home they also consider where they will store the items, how long each item will last, how quickly it will be used, and the family's attitude to brand. Cans are hefted, labels are examined, and the factors that influence shoppers have been made sufficiendy prominent by producers and retailers that shoppers can be certain to notice them. The effect is that reducing the problem of choice to comparing unit price strips the actual shopping problem of its complexity. Moreover, by placing competing brands side by side and placing related products nearby (spaghetti sauce near pasta), supermarkets provide a structure or organization for cognitive activity that biases they way shoppers think. Layout affects the way option sets are conceptualized (e.g., "I came in to buy spaghetti and decided to get linguini because I liked the look of the new Alfredo sauce"). This dynamic between product display and consumer framing of choice has coevolved.

In an earlier study, Carraher, Carraher, and Schliemann (1985) presented a related view. They found that Brazilian children selling goods in street markets invented special purpose procedures to add up prices and calculate change rather than use the more general pencil-and-paper methods they learned in school. They framed their problem in a domain-specific manner because the specialized cognitive artifacts they used to help them calculate were the ones built up in local practice and readily available in the situation.

For example, a girl who made money for the family as a street vendor, when asked the price of 10 coconuts selling at 35 cruzeiros per piece, did not use the add-0 method for multiplying by 10, as she had been taught in school. She used the cost of three coconuts (105), which was a convenient group she regularly sold coconuts in, added to this the cost of two more of these threesomes (210), then added the cost of a single coconut (35) to the running total. She correctly reported the price of 10 coconuts as 350 cruzeiros. Street market children did less well at these same problems in school, where they used the school-taught procedures. The authors noted that in the street, both children and older vendors used convenient groups for their additions, such as "three for 105," and that simple multiples of these groups, two or three of these three-for's were also very highly practiced so that, in effect, the vendors were substituting memory for summation or multiplication whenever they could. Predictably, school-taught procedures interfered with this type of situation-specific problem solving, and predictably, the street vendor kids performed better on the street than nonvendors with comparable education. Context and experience framed how the kids approached their problems and the resources and tools they deemed appropriate. Their activity in street environments was different than in classroom environments. Arguably, the cognitive resources in the street coevolved with the demands of street calculation.

In another study, this time by Scribner (1984), there is an account of how milkmen filled orders for different kinds of milk - white milk, chocolate milk, half-pints, quarts - by packing their delivery cases to make delivery more efficient and physically less effortful. This again is a numerical problem that was solved using contextualized knowledge.

Scribner noted that old-time milkmen used their delivery case itself as a thing to think with, and they filled orders faster and more accurately than students who filled orders using arithmetic calculations. The milkmen learned the numerical relations of various configurations of milk containers (one layer of half-pints is 16, two rows of quarts are 8, hence half as many as pints, and so forth) and used the compositional structure of layers and half layers, and so on, to fill cases with multiples of items without counting out each item. If they needed 35 half-pints to fill an order, they would know to fill two layers and then add three more on top. Deliverymen solved billing problems using a similar process of taking overlearned quantities or patterns, pulling them apart, and putting them back together. For instance, to figure out the cost of 98 half-pints a natural strategy was to take two times the case price (a case holds 48 half-pints and its price was memorized) and add the price of two half-pints. The patterns of milk cartons in the case are the elements of calculation. They became things to think with, patterns to decompose and recompose, to mentally manipulate. Again context and experience framed the way they saw their problems. The methods they developed were not universal, based on general algorithms for solving arithmetic problems; they were specialized and situation specific. And their deep familiarity with different situations showed in performance.

Cognitive scientists interested in learning theory have called the mathematical knowledge displayed here "intuitive" or "naive," to distinguish it from the formal knowledge taught in school (Hamberger, 1979). Intuitive knowledge is thought to be bound to the context in which the knower solves personally relevant problems.

The issue of who is right - the situational who looks at local resources as things to think with, or the formalist who frames the task more abstractly as a general type of problem, in these cases math problems that must be interpreted or applied to local conditions - lies at the heart of the situated challenge. Which problem are people trying to solve? If a subject thinks about a problem in concrete terms such as cans and shelves, and so has an internal conception and an external discourse that makes it seem as if the problem were about attributes of cans (e.g., their appearance, shape, volume, price, brand], why suppose he is deluded and is really talking about a basic number problem that happens to be couched in terms of cans? From his point of view, his problem is a naturally occurring one, quite unlike the contrived sentence problems presented in math class ("a bachelor comes to a supermarket with $15 looking for the best way to spend his money on pasta and beer. Spaghetti costs $1.75, fusilli cost $2.50, beer costs..."). From the formalist point of view, however, it is irrelevant that the numbers of interest refer to attributes of cans or bottles. Idiosyncrasies of the problem instance, such as what is near to what, or how information about price, volume, and so on, is displayed, do not matter. All that matters is the topological structure of the problem space, or the mathematical structure of the problem. And that may be the same whatever the labels are for nodes and links: can size, number of bottles, distances, or simply numbers.

Two findings discussed at length in the problem-solving literature - problem isomorphism and mental set - bear on this question of framing and abstraction. Both support the view that subjects are sensitive to surface attributes of a problem, so much so that two problems that are formally the same, or formally very similar, may be solved in such different ways and with such different speed-accuracy profiles that a process theory should treat them as different. Litde is gained by seeing them as only different problem-space representations of the same task environment.

Take problem isomorphism first. Tic-tac-toe and the game of fifteen are superficially different versions of the same problem (see Figure 15.2). Legal moves and solutions in tic-tac-toe and legal moves and solutions in the game of fifteen can be put in one-to-one correspondence. From a formal point of view the problems, therefore, are isomorphic. They have the same mathematical structure. Yet subjects conceptualize them differently and their performance is different, as measured by their speed-accuracy profiles and the pattern of errors they generate. Predictably, subjects rarely transfer their expert methods for tic-tac-toe to the game of fifteen; they relearn them. Evidendy, then, algorithms are sensitive to surface structure even if their success conditions are not.

What can be inferred? The obvious conclusion is that details of the problem context - the way it is presented and conceptualized, the richness of cues in the local environment - determine what subjects count as a solution and the resources they see as available to solve it. In the game of fifteen, if paper and pen are handy, for

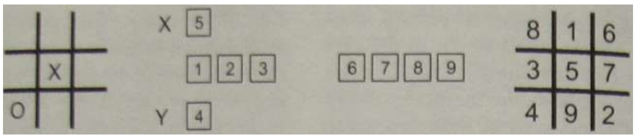


Figure 15.2. The task environment of the game of fifteen and tic-tac-toe is isomorphic because there is a one-to-one correspondence between the legal permissible moves in tic-tac-toe and the game of fifteen. In the game of fifteen, players take turns choosing from a set of numbered tiles. The first player to collect three tiles that sum to fifteen wins. Because it is easy to see opportunities for three in a row visually, but harder to see opportunities for summing to fifteen, subjects can play tic-tac-toe faster and with fewer errors than in fifteen. Their skills in tic-tac-toe do not transfer and they have to releam the tricks.

instance, subjects will often mark down sums and consequences of moves. How shall we view these paper actions? On the one hand, because actions on paper cannot improve the pragmatic position of a subject, paper and the actions it affords do not seem to be part of the problem context. On the other hand, for those who rely on paper to work out their next move, it is an important part of their problem-solving activity and makes a difference to their outcomes. In tic-tac-toe, scratch paper only gets in the way. Our visual system makes spotting consequences of moves easy. So in the game of fifteen versus tic-tac-toe, the resources, actions, and calculations relevant to a solution are, for many subjects, quite different. The formal state space of the two versions of the game is the same, but that space seems to abstract away from too many psychologically and activity-relevant details to explain the cognitive processes involved in problem solving. In fact, given such differences in problem-solving activity, why suppose the two even share an isomorphic task environment? The level of abstraction needed to view them in the same way seems too high. Because the purpose of a task-environment and problem-space approach is to provide us with constructs sufficient to explain psychological and behavioral activity, we need to find the right level of abstraction to capture generalizations. In these two cases, there seems too much difference in behavioral performance. Moreover, if our goal is a process model of problem-solving cognition, we ought to attend to the way subjects distribute relevant states over environmental artifacts (scrap paper as well as the spatial layout of cards or the way the tic-tac-toe board is filled in), and how they work out game moves by performing epistemic actions (Kirsh & Maglio, 1995).

Our concern here is with the possibility of finding the right level of abstraction to characterize the psychological processes involved in solving problems, even well-defined ones, such as the game of fifteen. In the gestalt theory of problem solving, it is assumed that people see a problem as a meaningful question only against a background of assumptions. To a given subject something is foregrounded as problematic only against this backdrop of the unproblematic (Luchins, 1942). The possible lines of solution that subjects will consider, accordingly, are constrained by their mental set, which limits the information they attend to and the conjectures and resources they think are relevant. Sometimes the mental set a person brings to a task or situation is appropriate and helps in finding a solution. Sometimes it does not.

An example of ways of framing and mental set can prevent problem solving is found in insight problems, where to solve the problem subjects must break out of conventional thinking and try something nonstandard, a trick. Usually this trick involves breaking preconceptions about what is allowable

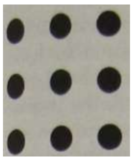
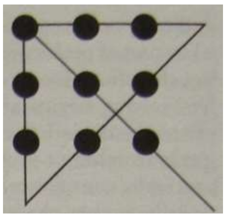



Figure 15.3. The nine-dot problem and its solution. The task is to connect all dots using four straight lines without lifting one's pen. Subjects frame the problem narrowly by assuming that lines must begin or end on dots and cannot extend beyond them. They incorrectly assume that all turning occurs on dots.

or what is the function of an available resource.

For instance, in the nine-dot problem, Maier (1930) told subjects to find a way to connect all the dots in a three-by-three matrix by using four straight lines, without lifting their pens or retracing any lines (see Figure 15.3). The problem is hard precisely because participants do not consider making non-dot turns (Kershaw, 2004). They rarely consider constructing lines that extend beyond the dots, and when they do they seldom consider making a turn in empty space, either between the dots or somewhere outside the matrix. The problem statement does not exclude these possible actions. But subjects frame the problem as if they consider these impermissible. Framing has prevented the subject from creating the right problem space, perhaps even from grasping the right task environment.

In separate work on mental set, in what is commonly regarded as the classical demonstration of set, Luchins (1942) presented subjects with water-jug problems in which they had to figure out how to get a certain amount of water (e.g., five cups), using any combination of three jugs: jug A holds eighteen cups, jug B holds forty-three cups, and jug C holds ten cups. They were free to dip their jugs into a well as many times as they like.

Luchins found that after subjects get the hang of the solution method and have more or less automatized it, they try to use that method on new problems and persevere in using it, even when there is a much better way to solve the new problems. This was seen as strong evidence for mental set because when other subjects were shown these alternative problems before the initial problem, they would soon learn the easy solution methods, suggesting that learning on one problem can interfere with learning and performance on another. Further support for the presence of mental set was found by Sternberg and Davidson (1983) and Gick and Holyoak (1980, 1983), who confirmed in new experiments that prior solution methods - prior set - worked against finding solutions to different problems, problems that subjects without that bias would be expected to solve.

The relevance of mental set for the situated approach is that how agents frame problems, what they see as possible actions, and good methods for success, depend on how they interpret their situation and their mindset in approaching a problem. To an experienced shopper, supermarket problems are their own sort of problem, quite unlike the general arithmetic problems learned in school. To the strongly mathematically inclined, however, supermarket problems are more likely seen as a special case of general arithmetic problems. Mathematicians see through the particulars of the shopping situation, grasping the more abstract mathematical problem. Their mental set is very different. And they worked hard to achieve that competence. To less mathematical reasoners, however, the supporting resources and scaffolds are so different and the tricks and visible cues are so different that, initially, at any rate, their whole mindset is different. The problem is different.

Even if one recognizes the abstract problem posed, the resources available still can strongly affect the method used to solve it. For instance, in math class at school students have pencil and paper. They write numbers down and rely on algorithms defined over the inscriptions they create. To multiply two numbers they line them up and use one of the multiplication algorithms. The same holds for division and determining ratios. Without pencil and paper, however, techniques and methods usually change. Even mathematicians might prefer to think with local artifacts if faced with a problem that is cumbersome to solve in their head.

The upshot is that though a task analysis may be important to determine the success conditions of different approaches, and indeed necessary to explain why they work, such analyses seem remote from a process theory. A psychological theory ought to explain the many phases and dynamics of the problem-solving process: how one sees a problem; why one sees it that way; and how one exploits resources, interacts with resources, and solves the problem in acceptable time. The bottom line, for the moment, is that how agents frame a problem, how they project meaning into a situation, determines the resources they see as relevant to its solution. If street vendors frame the "How much for ten of these?" problems in terms of today's price for three and today's price for one, then they prime a set of tools of thought distinct from those they learned in school. They do not look at the problem deeply. As work on transfer has shown, they stay on the surface, interpreting the problem in superficial ways.

2.2. *Registration*

As important as it is to extend the theory of problem solving to explain how problems are framed, this way of structuring problem solving still seems to locate the real part - the solving part - to take place after a task has been represented as a problem space; that heuristic search, in one form or another, is the driving force in problem solving, and that expertise is substantially about acquiring the right heuristics, metrics, and generalizations of cases as if framing is just a way of preparing for problem solving, not of solving it.

Two reasons to question this clean account are that first, creating a problem space may be a highly interactive process of framing, representing, exploring, reframing, and rerepresenting, so that reformulation is a key part of problem solving, and that framing, therefore, does not occur once and problem solving begins afterward; the two are often intertwined. Second, even when a subject is searching a problem space the search process is complicated by the need to continually anchor the search space in locally meaningful ways. Search itself is an interactive process that should not be reduced to internal symbol manipulation.

To appreciate these points, it is illuminating to contrast the concepts of registration and translation. In mapping a game of fifteen back into tic-tac-toe, we perform a translation. We similarly perform a translation when we map a word problem (Mary is two inches taller than Peter who is...) into a simple algebraic statement, or puzzles and games (nine dots, chess) into searchable graphs, or a problem in Euclidean geometry into a problem in analytic geometry using Cartesian coordinates. The value of the mapping is that the new representation offers another perspective with different methods and techniques, often simplifying problem solving. But the mapping process links two representations, or representational systems, and that is the key thing. Well-defined entities or relations in one representational system are mapped onto well-defined entities or relations in the other.

By contrast, when we orient and reorient a city or mall map to determine how the representations of buildings, pathways, and openings correspond with the buildings, pathways, and openings in the actual space, we are registering the map, not translating it, because we are trying to match up discrete representational elements in the two-dimensional map with nonrepresentational and often nondiscrete elements in the three-dimensional world, the arena where we perform physical actions. This means that much of problem-solving acumen, when registration is involved, may lie in knowing how to link representations (whether internal, external, or distributed over the two), with entities, attributes, and relations in the physical domain.

Examples of registration-heavy problems often arise when something goes wrong during practiced activity, when the normal method we use to get something done fails, and we are thrown into problem-solving mode to figure out how to recover. Cooking, cleaning, driving, shopping, assembly, and construction are all everyday domains where problems typically arise when there is a breakdown in normal activity. Much of what makes such problem solving hard is that the agent is not yet sure what to attend to: what events, structures, or processes to see as relevant. Every person has many frames for thinking about things, but which are the ones that fit the current situation?

Here is a trivial example, computer cases. Computer companies regularly devise new ways to open and close their cases and it can be surprisingly difficult to determine how to get inside a computer without first checking the manual. Problem solving consists of pushing or pulling on pieces, scrutinizing the case for telltale cues, for clear affordances or explicit indicators. It might be said that this activity is a form of external search. But more likely it is a form of registration: of trying to discover a pattern of cues that can be fit to a method we already know or to a mechanical frame that will make sense of the release mechanism. This is a form of registering because much of the reasoning involved in determining how to open the case is tied to exploring the affordances of the object and looking for ways to conceptualize or reconceptualize its different parts. In fact, in many everyday problems, the registration phase is more complex than the search phase.

In some instances, the phases of registration and search are virtually impossible to separate. We can rationally reconstruct problem solving so that there are distinct phases, but in fact the actual process is more interactive. This is especially trueofwayfinding with a map. We can, if we like, describe map use sequentially: first, orient the map with immediate landmarks to establish a correspondence; second, determine current position; third, plot a route; then, fourth, follow it. But discovering and following a route is typically interactive: look at the map, look at the surroundings, locate oneself, interpret map actions in physical terms, and repeatedly do this until the goal is in sight.

Navigational capacities depend on continually linking symbolic elements in the map to physical referents in the space. These referents serve as anchors tying the map down to the world so that a trajectory in the map can be interpreted in terms of visible structures in space. Thus, in a shopping mall we look for signs and arrows pointing to the food court or stores of interest to help us figure out where we are. We interactively make our way, often by working off the physical setting rather than the map. But when plotting a course we use all these cues to help us orient and make sense of the path we devised using the map. Subjects go back and forth between map and world. The reason this constant anchoring has not been a major issue is that in games such as chess, Tower of Hanoi, and tic-tac-toe, in math problems, and other verbally stated problems, interaction is focused on a spatially constrained representation, the chess board, the tower, the formulation of the problem. This sustains the dlusion that problem solving is primarily a matter of controlling operators in an abstract representation.

The upshot is that in many naturally arising problems the locus of difficulty may lie as much in the registration process, the activity of selecting environmental anchors to tie mental or physical representations to the world, as it does in searching for paths in the representation itself.

3. Interactivity and Epistemic Actions

3.1. *People Solve Problems Interactively*

In most problem-solving situations people do not sit quiedy until they have an answer and then announce it all at once. They do things along the way. If it is a word problem (John is half as tall as Mary...), they mutter, they write things down, and they check the question several times. If they are solving an assembly task (here are the parts of a bicycle, assemble it), they will typically feel the pieces, try out trial assemblies, and incrementally work toward a solution. Rarely does anyone work out a complete solution in their head and then singlemindedly execute it. People, like most other creatures, solve things interactively in the world.

The classical approach to problem solving failed to adequately accommodate this in-the-world and not just in-the-head interactivity in two ways. First, the classical theory, in its strictest form, assumed that users completely search an internal representation of their problem before acting. Heuristics were proposed as a mechanism for reducing the complexity of this internal search so that solutions could actually be found. They were not meant to help a user figure out the next single action to perform; they were meant to help a user figure out a whole plan, an entire sequence of actions. This "Make a plan before you act" hypothesis was derived from Miller, Galanter, and Pribram (1960), and not surprisingly was repeatedly challenged in the planning literature, both in AI and in psychology.

The second way interactivity was misunderstood was that it was never seen as a force for reshaping either the search process or the problem space. Artificial intelligence theorists were quick to appreciate the value of incorporating sensing and perception into planning. But most AI planners incorrectly assumed that any actions that users perform in the world during problem solving are either

- External analogues of internal search - instances of searching in the world instead of in the head, or
- Stepwise execution of an incomplete plan - starting to implement a partial solution (or plan) before having the complete one in mind - then replanning in light of the resultant world state.

External interactivity was never (or rarely) seen as a mechanism for reducing the complexity of a problem, or as a mechanism for exploring the structure of a problem or as a way of engaging other sorts of behaviors that might help subjects solve their problems. External activity was still related to search, one way or another. A theory of situated problem solving should give the principles of more interactive approaches.

3.2. *The Role of External Representations*

Although Simon and other exponents of the classical theory never accepted the centrality of interactivity, they took an important step forward when they began paying more attention to the role external representations play. Larken and Simon (1987) enlarged the orthodox account to allow problem states to be partially encoded internally and partially encoded externally. Accordingly, to solve a geometric or algebraic problem, subjects might rely on applying operators to external symbols, equations, illustrations of geometric figures, and so forth. Instead of representing the transformations of the equation $2x + 4y = 40$ in one's head, as in mental representations for $2(x + 2y) = 40$ followed by $x + 2y = 20$, Larken and Simon showed that it might be easier to generate such representational states in the world and track where one is both mentally and physically to decide what to do next.

The special value of external representations is obvious in visual problems, such as tic-tac-toe. Vision is a computational problem that terrestrial animals devote huge neural resources to solve. In tic-tac-toe, the computational cost of evaluating the consequences of a move can be borne by the visual system, which exploits parallel and highly efficient methods to project the outcomes of placements. This makes it easy to see that one or another move is pointless. For instance, in Figure 15.4a, given a visual display of the board, it is obvious where o must move to prevent immediate defeat. Choice can be made without serious consideration of other moves. Solving the same problem algebraically, as in the game of fifteen, or solving the same problem in one s head, especially more complex versions (see

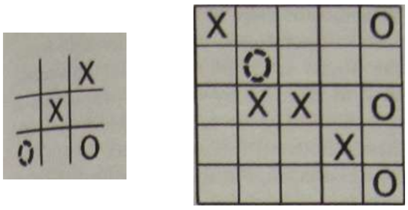


Figure 15.4. In harder versions of tic-tac-toe it is nearly impossible to determine the best moves without a visual representation of the board. Although the board state of 15.4a can be generated internally, and players can easily play without looking at the board, the cost of sustaining such states in mind increases with the complexity of the mental image or representation. In 15.4b, the five-by-five board is much less easy to keep in mind. To win it is necessary to get four in a row anywhere. As the structure of the problem increases and the complexity of the current state rises, vision pays off. The interactive nature of vision scales better with board complexity.

Figure 15.4b) is significantly harder. It is both cognitively easier and computationally simpler to use the external representation than an internal representation.

In treating problem solving as a process that may be partly in the mind and partly in the world, the classical view took a big step toward a more situated perspective. But the assumed value of external representations lay in the increased efficiency of applying visual operators and the stimulating role external representations can play in search. Using external representations was not seen as forcing a revision of the way problem solving unfolds. In particular, a concern with external representations did not lead to a discussion of other ways external resources figure in problem solving. Let us consider the role of external representations more closely.

Expanding the search space. In an elegant demonstration, Chambers and Riesberg (1985) showed that how people visually explore an external representation is often different from how they explore a mental image of the same thing (see Figure 15.5).

If you look at a Necker cube for half a minute or more your interpretation will almost certainly toggle, and the surfaces you see as front and back will swap places. That is, if you first see A as the front face and B as the back, then after a short while you will see B as the front and A as the back. This swapping of faces, this reinterpretation of the figure, does not occur in mental images. A mental image is an intentional object and as such must be sustained under an interpretation. If a mental image were conceptualized as an organized set of lines not yet interpreted as a cube, then new mental images might arise from thinking about the image. But if it is maintained as a unified object - a gestalt - it will not toggle unless deconstructed into constituent lines. People rarely deconstruct their mental images.

What if the same cognitive limitations apply to internal problem spaces? What if visual operators can explore parts of a search space more broadly than internal operators acting on mental representations of the same structure? This would suggest that certain problems might be solved only when they

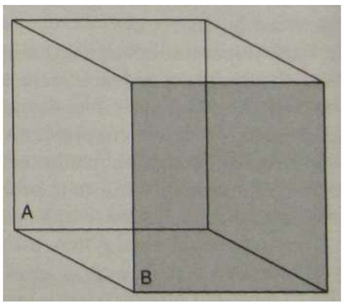


Figure 15.5. The Necker cube and other visually ambiguous structures have more than one interpretation, which subjects discover after a short while. When the cube is visually before them, they scan the edges, propagating constraints. But when the same image is imagined, the ambiguity of the figure goes undetected. It is grasped as a whole, without the need of "mental saccades" to test its integrity and sustain it.

are represented externally in figures or symbols. This would be a curious form of cognitive set.

In some of his work, Simon came close to making this point. In several coauthored articles he considered the role that auxiliary representations play in helping subjects solve algebraic word problems. He found that students used diagrams to detect additional assumptions about the problem situation that were not obvious from the initial algebraic encoding of word problems (Simon, 1979). Although such additional assumptions could in principle be discovered from the algebraic encoding alone, subjects found the cost of elaboration too great. Discovering such assumptions without diagrams requires far too much inference.

Given the interest in problem isomorphs at the time, proponents of the (revised) classical view did not regard the value of external representations as grounds for shifting the focus of problem-solving research away from heuristic search in problem spaces, to replace it with a study of how subjects interactively engage external resources. It was instead seen as further evidence that how subjects interpret problems affects their problem-solving behavior. It nicely fit the problem isomorph literature showing that surface representation strongly determines problem-solving trajectory.

In some respects it is surprising that the classical theory did not embrace interactivity at this point. The idea that it is useful, and at times necessary, to transform problems into different representational notations or representational systems is a truism in problem solving in math, and in science more generally. Every representational system makes it easy to represent certain facts or ideas and harder to represent others. What is explicit in one representation may be implicit in another (Kirsh, 1991, 2003)1 For example, in decimal notation it is trivial to determine that 100 is divisible by 10. But when 100 is represented in binary notation as 110010 it is no longer trivial. This holds whether we translate 10 into its binary equivalent, 101, or keep it in decimal notation. The decimal notation is better than binary for certain operations, such as dividing by 1-the binary is simpler for other operations such as dividing by 64. The proponents of the classical view certainly knew this and often discussed the importance of problem representation, but they never took the next step.

Their appreciation of the importance of diagrams, illustrations, and word formulations for problem-solving performance never led to a major departure from the problem-space, task-environment idea. Externalization was not seen as establishing a need to shift focus from problem spaces to affordances or to the cues and constraints of external structures. It never led to a revised concern for observing what people actually do, in an ethnographic sense, when they solve problems.

3.3. *Adding Structure to the Environment*

When we do look closely at the range of activities that people perform during the course of solving or attempting to solve problems, we find many things that do not neatly fit the model of search in an internal or external problem space.

For instance, in their account of subjects playing the computer game Pengo, Agre and Chapman (1987) discussed how a computer program, and by analogy humans, could exhibit planned behavior without search. Their program worked interactively. It used a set of simple rules to categorize the environment in a highly context-sensitive manner. The environment that *Pengi* - the name of the computerized penguin that Pengo players were attempting to control - consists of blue ice-blocks distributed in a random maze. Pengi starts in the center and the villains of the game, the sno-bees, setoff from the corners. If Pengi is stung it dies, but it can defend itself by kicking an ice-block directly in front of a sno-bee thereby crushing it. If Pengi was chasing a bee, Agre and Chapman's system would classify the structural arrangement of the ice-blocks one way. If Pengi was running away from a bee, the system would classify the very same arrangement of ice-blocks another way. To achieve this difference in classification the system attached visual markers - visual memory projections or annotations - to certain parts of the situation. Thus, on two occasions, the same objective state might be classified in two ways depending on which of Pengi's goals were active, because the world would be visually annotated differently (see Figure 15.6).

Agre and Chapman then showed that with these visual markers the computer program was able to behave in a strategic manner using a few reactive rules. It was not necessary to search a problem space as long as the system could project additional representational structure onto the visible environment. The implication was that humans work this way, too. By performing certain types of visual actions, including actions that affected visual memory, humans are able to solve problems without search that are both complex and that on a priori grounds ought to require extensive search.

This tactic of adding structure, either material or mental, to the environment to simplify problem solving is surprisingly pervasive. People mentally enrich their situations in all manner of ways. To help improve recall there are strategies such as the method of loci, which involves associating memory items with spatial positions or well-known objects in one's environment. To improve performance in geometric problem solving, people project constructions, mental annotations (see Figure 15.7).

People have even more diverse ways of materially enriching their situations. They add reminders, perhaps with Post-it Notes, perhaps by rearranging the layout of books, papers, desktop icons, and so forth. They annotate in pen or colored pencil; they encode plan fragments in layouts; they keep recipes open; and of course they talk with one another, often asking for help or to force themselves to articulate their ideas, using their voice as an externalized thought. Indeed, it is widely accepted that the act of collecting one's thoughts to present a problem to another person is an effective method to identify and clarify givens, to articulate problem requirements, and to expose constraints on problem solutions or solution paths (Brown, Collins, & Duguid, 1989). Some of this facilitation, no doubt, occurs because different representations elevate different aspects of a problem. But some of it, as well, is due to the known value of talking out loud during problem solving (Behrend, Rosengren, & Perlmutter, 1989).

The thread common to all these different actions is that they reduce the complexity of the momentary computational problems that agents face. They help creatures with limited cognitive resources perform at a higher level.

Here is another, more prosaic example. In a card game, such as gin rummy, players tend to reorganize their hand as they play. Reorganizing a hand cannot change the value of current cards or the value of subsequent cards. Whether or not an ace of spades will be a good card to accept and the three of clubs a good card to throw away is unaffected by the way players lay out the cards in their hand. The objective problem state of the hand is invariant across rearrangement. Yet from a psychological perspective, rearrangements help players to notice possible continuations and to keep track of plans. Thus, the strategy ***sort by suit then sort in ascending order across suit,*** is an effective way to overcome cognitive set, or continuation blindness. This simple interactive procedure effectively highlights possible groupings. It is an epistemic activity. The knowledge dividend it pays exceeds its cost to an agent in terms of time and effort (Kirsh, 1995b).

3.4. *Epistemic Actions*

In a series of papers Kirsh (1995a, 1995b; Kirsh 8t Maglio, 1995) have argued that this sort of epistemic activity is far more prevalent than one might expect. Even in contexts where agents must respond very quickly it still may be worth their while to perform epistemic actions. For instance, in the arcade game Tetris, the problem facing players is to

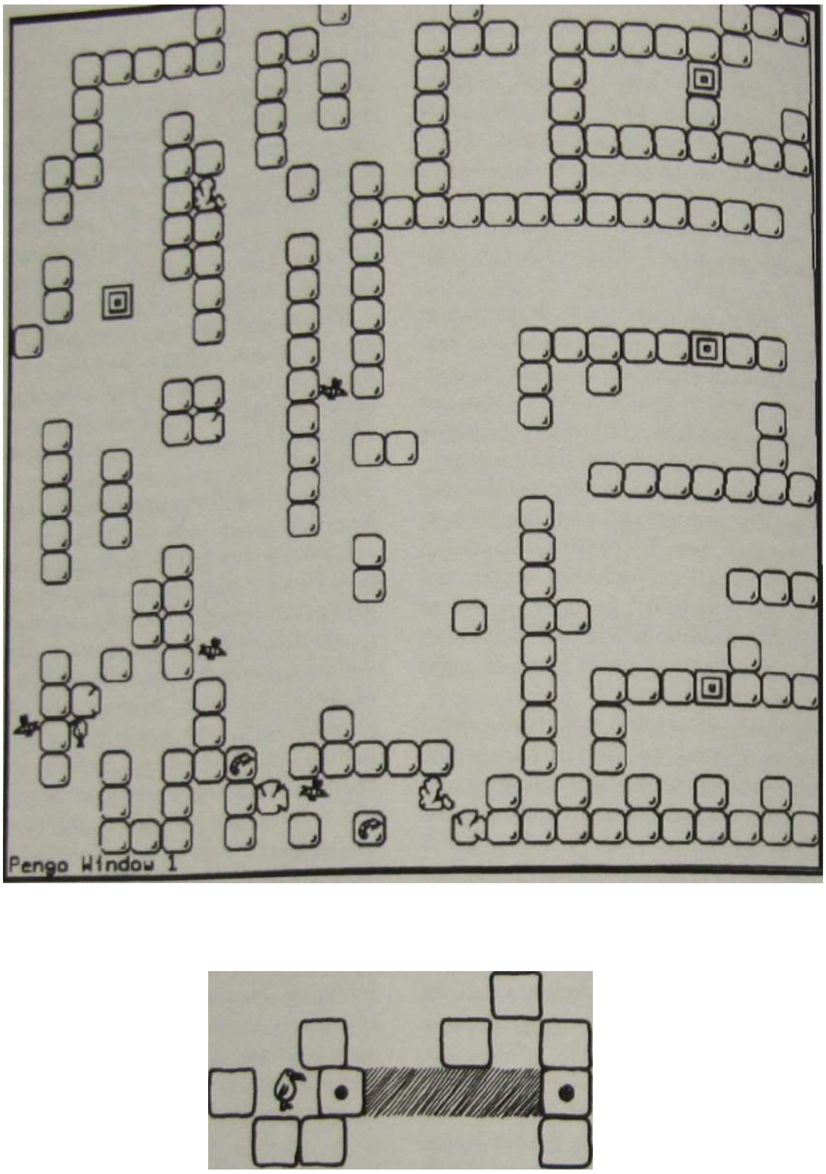


Figure 15.6. Here we see the game board of Pengo as a human sees it and a blowup of one section of a game where Pengi projects visual markers, indicated by dots, to allow it to act as if it has a rule "Kick the 'block-in-front-of-me' to the 'block-in-its-path.'" The dot markers serve as indexical elements that Pengi can project so that its current visual working memory has enough structure to drive the appropriate reactive or interactive rules it relies on to determine how to act. Some of these rules tell Pengi to add visual structure and others to physically act.

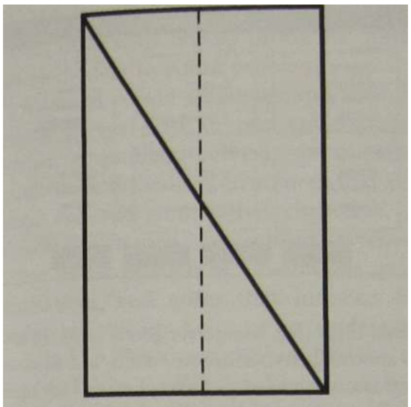


Figure 15.7. Humans can readily add mental structure to an illustration by projecting lines, bisecting angles, and generally adding mental annotations. In this example, an agent is shown a rectangle and asked to prove that a line that cuts a rectangle in half will also cut all diagonals of the rectangle in half. In those cases where subjects do not draw in the diagonal and bisector, they still can often find a solution by projecting first a diagonal and then a section line that bisects the rectangle. It is possible to add letters to projected points such as the intersection of the dotted bisection and the solid diagonal. This same capacity can be used in creative ways to add structure to other sorts of situations by envisioning the results of performing actions before performing them.

decide how to place small tetrazoidal shapes on a contour at the bottom of the board (see Figure 15.8). As the game speeds up, it becomes harder for players both to decide where to put the shapes and to manipulate them via a keyboard to put them into place.

What Kirsh and Maglio (1995) found was that players, even expert players, regularly performed actions that helped them to recognize pieces, verify the goodness of potential placements, and test plans (e.g., dropping a piece from high up on the board) despite there being a cost to their actions in terms of superfluous moves. Evidently the cost of moving a piece off its optimal trajectory was more than compensated for by the benefits of simplifying some aspect of the cognitive problem involved in identifying the piece and determining its best resting place. The novel feature of these epistemic actions is that their value depends crucially on when they are done. Because the game is fast paced, information becomes stale quickly. Twirling a "zoid" the moment it enters the board is a valuable action, but it is near useless to an expert 200 ms later. Twirling must be timed to deliver information exactly when it will be useful for an internal computation. From a purely problem-space perspective, where states and operators have a timeless validity, there is no room to explain these time-bound actions. Even though epistemic actions help agents to solve problems, and they can be understood as facilitating search by increasing the speed at which a correct problem representation can be created, they are not, on the classical view, part of problem solving, and they do not lie on a solution path. They help to discover solutions, but for some reason they are not part of the solution path.

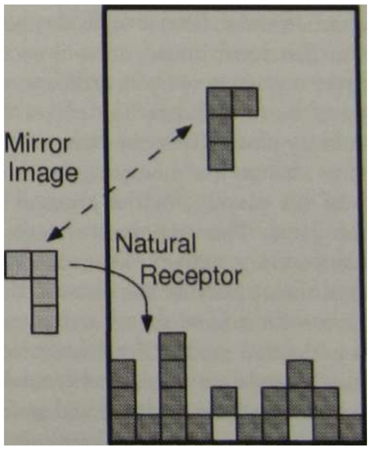
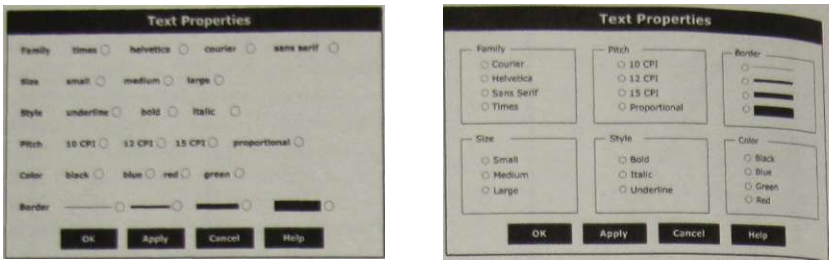


Figure 15.8. In Tetris the goal of play is to relentlessly fill gaps on the bottom layers so as to complete rows. The game ends when the board clogs up and no more pieces can enter. Because there are mirror pieces, and the choice of where to place a Tetris piece is strategic, players need many hours of practice to become expert. We found that players often perform unnecessary rotations to speed up their identification process, especially among pieces with mirror counterparts. (Kirsh & Maglio, 1995)



a b

Figure 15.9. The consequence of thoughtful redesign is that tasks that were error prone and hard to manage become progressively easier. The two interfaces are small environments - tools, in a sense - that agents must thoughtfully use to complete a task. In this case the task is to set the font style of a region of text. By redoing the graphic layout, as in 15.9b, designers are able to lower the costs of planning what to do, monitoring what one is doing, and verifying that one has completed the task. If this task was treated as a problem to solve, the effect of redesign is that it is now easier, even though the task environment is the same.

The challenge epistemic actions pose to the classical approach, and to the psychological study ofbehavior more generally, is that without an analysis of the possible epistemic functions of an action it may be nearly impossible to identify the primary function of an action and so label it correctly. Actions that at first seem pragmatic, and so to an observer may seem to be an ordinary move, may not be intended by the player to be an ordinary move. Their objective may have been to change the momentary epistemic state of the player, not the physical state of the game. This important category of problem-solving activity lies outside the classical theory because the classical theory operates with a fixed set of actions and a fixed evaluation metric. Both action repertoire and metric are taken as objective features of the task environment and assumed insensitive to the momentary epistemic state of the agent.

4. Resources and Scaffolds

The classical theory also fails to accommodate the universality of cultural products that facilitate activity-specific reasoning. Environments in which people regularly act are laden with mental aids, so problem solving is more a matter of using those aids effectively. Supermarkets tend to display competitive items beside one another to help shoppers choose; large buildings tend to have signs and arrows showing what lies down a corridor because people need to find their way; microwaves and ovens have buttons and lights that suggest what their function is because people need to know what their options are; and most of our assembly tasks take place when we have diagrams and instructions. Wherever we go, we are sure to find artifacts to help us. Rarely do we face knowledge-lean problems like the Tower of Hanoi or desert-island problems. Our problems arise in socially orga nized activities in which our decisions am activity are supported.

There are several reasons why cultun products and artifacts saturate everyda environments. First, most of the enviror ments we act in have been adapted to hel us. Good designers intentionally modify ei vironments to provide problem solvers wi' more and better scaffolds and resourcf designs that make it easier to complete tas (see Figure 15.9 and caption). And if desig ers are not the cause of redesign, it oft happens anyway, as a side effect of previc agents leaving behind useful resources after having dealt with similar problems. Rarely are we the first to visit a problem.

A second reason resources and scaffolds almost always exist in our environments is that, as problem solvers, we ourselves construct intermediate structures that promote our own interactive cognition. As already mentioned, we create problem-aiding resources such as illustrations, piles, annotations, and notes that function like cognitive tools to help us to understand and explore questions and coordinate our inquiry. When we face problems, we typically have a host of basic resources at hand: tools such as pen, paper, calculators, and rulers; manipulables such as cans, cups, and chopping boards; cultural norms of gesture, style, language; and, of course, cultural resources and practices such as tricks for solving problems, techniques, algorithms, methods of illustration, note taking, and so on. When we attempt to solve a problem, we reach for these aids or call on tricks and techniques we have learned. Many of our solutions or intermediate steps toward solution take the form of actions on or with these materials. They seed the environment with useful elements (e.g., lemmas), they make it possible to see patterns or see continuations (e.g., the lines in tic-tac-toe), or they make it easier to follow rules (e.g., the lines in multiplication; see Figure 15.10).

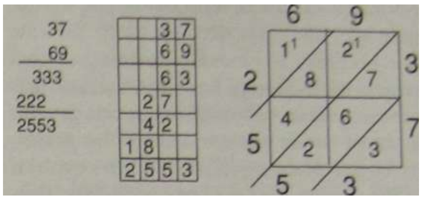


Figure 15.10. Multiplication problems are rarely solved in the head. Problem solvers reach for pen and paper, line up the numbers in a strict manner, and then produce intermediate products that are then relied on in their algorithm. Paper, pen, graph paper, and lines are all scaffolds that help agents to reduce error. In this figure we see three ways to multiply. The most noteworthy feature of these structures is the lines that guide the user. They are not elements of the algorithm. They are scaffolds indicating where the multiplicands end and the intermediate products of the problem solver are to be placed, and so on. They help agents stay in control, keep on track, monitor where they are, prevent error.

A final source of resources and scaffolds is found in our neighbors or colleagues who are often willing to give us a helping hand, offering hints, suggestions, tools, and so on. In educational theory, the term ***scaffolding*** refers to the personalized problem-solving support that an expert provides a novice. ***Help*** is too simple a term to describe this support because a good teacher gives a student tools, methods or method fragments, and tricks about technique that the student is ready to absorb but that, taken on their own, do not provide an answer to the student's impasse. Scaffolds extend the reach of a student, but only if the student is in a position to make use of them. This was a key aspect of the way Vygotsky, the author of the term, used it. In his view scaffolds are support structures in learners' immediate environment that might permit them to solve problems that are at the periphery of their problem-solving ability, problems that reside in what he called the "zone of proximal development." They are akin to hints. And intrinsic to this notion is the assumption that as soon as the student internalizes the requisite methods, norms, heuristics, and construction skills, scaffolding will be unnecessary and no longer found in the problem-solving situation. Workers put up scaffolds to help them reach parts of a building they cannot otherwise reach. But as soon as they have finished their job, they remove the scaffold. Training wheels are a classical example of scaffold in the learning literature. Other examples include the use of vowel markers in beginners' Hebrew that are omitted in normal Hebrew writing because context and knowledge make them redundant.

Outside of learning theory, the term ***scaffold*** is used to refer to the cultural resources - artifacts, representations, norms, policies, and practices - that saturate our everyday work environments and that remain even after we have internalized their function. They reflect the supports present in most work environments. The major-it)' of our problem-solving abilities evolved in these resource-heavy environments; they rely on those resources being there.

Take the case of geometric problems. Students leam a variety of ways to solve such problems but most involve constructing illustrations. Part of a student's problem-solving competence consists in the ability to create apt illustrations and then to use them, not just to understand the problem but to work inside, annotating them, to solve the problem. In Figure 15.11, we see an example. Rather than tackle the problem algebraically, an easier approach is to modify the problem-solving environment so that it supports a range of different actions that the user finds easier to control and work with. As discussed earlier, these representations have a structure that makes the relevant attributes of the problem easier to manage.

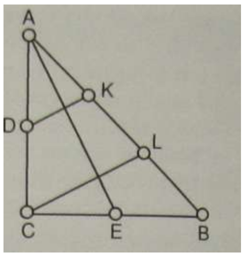


Figure 15.11. The written or formal statement of a geometric problem is almost impossible to understand without translating it first to a geometric diagram. In proving KL == BL, virtually all problem solvers use the figure as an arbitrary version of the problem that feeds intuitions. Students find it simpler to reason using the figure because it is easier to notice visual or geometric relations between properties in this problem than it is to notice algebraic relations. The validity of the proof depends on universal generalization: a proof method where one assumes an entity, an isosceles triangle of certain dimensions; one then proves a key statement about this triangle; and then one shows that nothing in the proof depended on having chosen this particular triangle as the example. The proof generalizes to all right-angled isosceles triangles.

> Formal Problem. ABC is an isosceles right-angled triangle with the right angle at C. Points D and E, equidistant from C, are chosen arbitrarily on AC and BC. Line segments from D and C are perpendicular to AE and meet the hypotenuse AB at K and L. Prove that KL = BL.

The presence of resources and scaffolds in an environment does not, in itself, challenge the view that people solve problems by searching through a problem space that is distributed over structures in the world and structures in the head. But it does call into question how to formulate the problem that people are solving. If the resources at hand change, or are different from situation to situation, why assume the problem is the same? Does a student who uses algebraic techniques solve the same geometric problem as his friend who uses an illustration? Do they operate in the same task environment? A task environment, after all, is defined with respect to operators too. How about a student who uses a pocket calculator to determine the square root of 1600 versus her friend, who calculates the value by hand: is it the same task environment? Tools, scaffolds, and resources seem to interact with tasks, usually changing them, often in ways that reduce their complexity.

5. Special-Case Solutions

A favorite pastime of situationalists is to enumerate instances in which resources in the environment have been put to creative use to transform the complexity of problem solving. No discussion of situated problem solving would be complete without mentioning the infamous example, "The intelligence is in the cottage cheese" (see Figure 15.12)-

Although the method shown in Figure 15.12 is certainly simple, it is worth noting just how narrow it is. It is only because the daily allotment of cottage cheese is 3/4 of a cup, and tubs are exactly 1 cup (or because consumers have a measuring cup on hand), that it is possible to execute the algorithm and use the visual cues provided by the crisscrossing. This is a very special case. Weight watchers' dieticians might have allowed 7/16 of the daily portion of cottage cheese as the lunch portion. And they might have decided that the daily allotment was 3/5 of a cup.

But why would they? Problems and the environments in which they are solved are rarely independent. If for some reason 21/80 of a cup became important among a subset of consumers, how long would it be before manufacturers would change the tub size to make such calculations easy?

This highlights a key fact about situated problem solving: it is usually narrow, special-case-oriented, and shallow. The reason such approaches work is that most instances of naturally occurring problems, even problems that are theoretically hard, are confined to those versions of the problem that can be solved easily. The ones that resist simple methods are part of a small set of worst-case versions (technically known as the complexity core) that people rarely if ever encounter. On those improbable occasions where they do encounter an instance of this core, they usually do poorly.

Here is an example. The traveling salesman problem is computationally intractable in the general case. There exist some nasty problem instances - the worst cases, the complexity core - where the only way to determine an optimal tour among all the cities the salesman must visit is by checking every conceivable tour. Because a tour consists of visiting every destination exactly once, there are as many tours as sequences of destinations, and that is *n* factorial tours. For ten cities that means checking 10! or 3,628,800 tours.

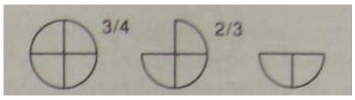


Figure 15.12. To solve the problem of how much cottage cheese two-thirds of three-quarters of a cup is, a subject was observed to turn over a one-cup tub of cottage cheese, crisscross it to mark four quarters, remove one quarter, then take two of those three quarters, which is one-half. It would have been easy enough to multiply 2/3 by 3/4, but typical subjects require pencil and paper to do that and regard the task to be effortful and confusing. The cup-sized cottage cheese itself became a thing to think with.

Yet in practice, most of the problem instances a traveling salesman actually faces are not nasty. Indeed, depending on the road layout of a given sales region, it may be quite easy to compute an optimal tour (see Figure 15.13). Would a salesman well adapted to his specific region leam a slow general algorithm for solving the problem in all cases or a fast special algorithm that satisfactorily solves just his customary problems? Who would be better adapted to his task? And, if there happens to be one or two worst-case problems lurking in his sales region where his method gives him a suboptimal tour, the overall cost of traveling that imperfect tour one day a month is more than compensated by the benefits of having cheaply found optimal or near optimal tours the rest of the times he has gone out.

Findings in complexity theory justify this perspective. Hard problems are almost always easier if a second-best answer will be adequate much of the time. More precisely, a problem that is NP-complete (or worse) can usually be solved in polynomial time if it is acceptable to give a solution that is 1 - 6 from the perfect answer (Padadimitriou & Yannakakis, 1988).[1] The more tolerant one is about the precision of the answer - that is, the further from the perfect answer one can tolerate, the faster the algorithm. So if our traveling salesman does not require having the absolutely shortest path but will

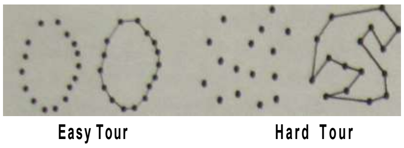


Figure 15.13. Compare these two instances of the traveling salesperson problem. Why is the easy tour easy to compute and the hard tour hard? Because the easy tour lies on a convex hull. If we know this fact, then we can use a fast algorithm based on the truth of that assumption. If there is no convex hull, or if we have no reason to think that there is, then there is no fast algorithm to guarantee an optimal tour. In many cities, roads are laid out on a grid structure, which makes the determination of optimal tours more like easy than hard tours.

settle for one that he knows is close to the shortest, then he can have a quick, reliable method of finding that path. And the same benefits apply if he can tolerate a little uncertainty in his answer. Thus if he can accept being right at least 99 percent of the time, or more precisely, have confidence 1 - E that his answer is optimal, then a fast algorithm exists (Karp, 1986). The reason such algorithms exist is that in all problems there is a large class of instances where there is structure in the problem that can be exploited. In Figure 15.13, the structures are apparent. In Figure 15.12, the cottage cheese case, the structure is the obvious relation between the numerator and denominator of the fractions 2/3 x 3/4, 1/2 x 2/3, 3/4 x 4/5, and so on. An agent does not have to know about this structure or understand it to rely on algorithms or heuristic methods that are valid in virtue of this structure. But an agent who is ignorant of the relevance of this structure will never know whether she is currendy confronting an instance of the hard core or why her method sometimes fails.

The adequacy of special-case solutions suggests that, in general, agents operate with an understanding of their problems that is good enough for the cases they normally encounter. They conceptualize their context rather differendy than formalists, who see the problem being solved as an instance of a more general mathematical, logical o planning problem. This raises a hard prob-lem for formalists. How should the conceptualization people actually work with - the conceptualization that somehow figures in the problem space an agent creates - be matched against the real task environment associated with the problem in its more general form?

The answer to this question has deep implications for problem-solving methodology. As empiricists we ought to accept that it is an empirical question which of the many possible task environments that might fit a problem is the actual task environment that problem solvers must adapt to. It is not to be answered in ignorance of the frequency of the problem instances real problem solvere face. Presumably, it is misleading to choose too general a task environment if, in fact problem solvers never face more than a small subset of the problem instances that fit the general task specification. That would be a bad framing of the problem.

For example, the task environment for the Tower of Hanoi, in most renderings of the task, is structurally the same whether it applies to towers made with thirty disks or to those made with three disks. From a mathematical point of view the task environment is the same because a recursive algorithm is used to generate the formal structure of the problem. Just as we treat multiplication of two thirty-digit numbers to be the same basic task as multiplication of two two-digit numbers, so it might seem natural to treat solving thirty-disk problems to be the same basic task as solving three-disk problems. Of course the actual problem space is much larger in the thirty-disk case than in the three-disk case because there are so many more combinations to consider. But why should that matter? If we think that subjects solve Tower of Hanoi problems with a recursive algorithm, it ought to be the same whether they face thirty disks or three.

Yet people who solve the three-disk problem often fail on larger problems, even ones as small as seven- or eight-disk problems. They usually have trouble keeping track of where they are in the problem; they have trouble maintaining current state. At some point they need a separate strategy to stay on course, and typically this is to use paper or some other way of encoding state. Can we pretend that this auxiliary strategy is not part of their problem-solving method for larger problems?

This raises the question: Do they face the same problem in both cases but use different methods because of memory and computational limits? Or do they face different problems because small and large versions of the Tower of Hanoi are actually different problems given the resources and methods subjects have? The answer depends on why we think it useful to invoke a task environment. The notion of a task environment was introduced to explain what rational agents adapt to as they get better at solving a problem. Namely, they adapt to the problem's structure, to the cues and constraints on paths to a solution. The paradox of large problems is that rational agents never do adapt. They cannot use the algorithms they use for smaller instances of the "same" problem, at least not in their head. And when they use paper, they perform many other actions related to managing their inscriptions that have nothing to do with the core algorithm. Why, then, maintain that the language of task environments is helpful? What does it predict? It does not predict the performance and pattern of errors that problem solvers display as the problem gets larger, because performance depends on the algorithms actually in use. All it can predict is how an ideally rational agent, unaffected by resource considerations, would behave.

If it is true that people do not use the same method in large and small versions of the Tower of Hanoi, why assume that they should use the same methods in other environments, such as shopping, assembly, selling, or navigation?

The upshot is that situated problem solving emphasizes that people solve problems in specific contexts. The methods they have learned are well adapted or efficient in those contexts but may be limited to special cases, not generalizable, and often idiosyncratic. Indeed, given the coevolution of settings and methods of coping, we would expect that most problem solving in well-designed environments is computationally easy, with external supports that ensure it is so. No one has argued that situated problem solving is better than other methods. It is just more like the way we think. And that was the question at issue.

6. Knowledge-Rich Cognition

Experts know a lot about their domains. Even if they cannot articulate their knowledge, they have built up methods for achieving their goals, dealing with hassles and breakdowns, finding work-arounds, and more to make them effective at their tasks. That is why they are experts.

In regarding agents in their everyday contexts to be well adapted to their contexts of work and activity, we are treating them as rich in knowledge, as more or less experts in their commonsense world. They know how to get by using the material and symbolic resources supporting action. Perhaps the difference between situated cognition and problem-space cognition is the difference between knowledge-rich and knowledge-lean cognition.

An old distinction, still useful but potentially misleading, distinguishes declarative from procedural knowledge: knowledge of facts and explicit rules from the type of knowledge displayed in skills. Typical studies in knowledge-rich problem solving focus on the dense matrix of facts, procedures, heuristics, representational methods, and cases that experienced agents bring to their tasks. This is a mixture of declarative and procedural knowledge, though presumably some of the knowledge that experts have is related to knowing how to publicly use representations, to exploit their social networks, and to use tools - skills that seem to be more embodied than can be encapsulated in a set of procedures. Because most people become experts or near experts in dealing with their everyday environments - shopping, driving, socially conversing, preparing their meals, coping with familiar technology - they probably know enough about these domains to have effective problem-solving methods for handling the majority of problems they confront. For the few problems they cannot handle, they usually have work-arounds, such as calling friends for advice or knowing how to halt or abort a process, that let them prevent catastrophic failure.

On the classical account, a major source of the improved performance that experts display is to be explained in terms of improved search or improved representation of problems. They have better methods for generating candidate solutions and better metrics for evaluating how good those candidates are. It has also been observed that experts spend more time than novices in the early phases of problem solving, such as determining an appropriate representation of a problem. To cite one study (Lesgold, 1988), when a fund manager considers stocks to include and exclude from her portfolio, she is systematically reviewing a set of candidates. So we might start our interpretation of her problem solving with a problem space and operators. But before she makes her final decision she will have done many things, perhaps over several days, to uncover more information about stocks, and to get hints about what the Street thinks about each stock. Some of the things she might do include retrieving charts, extending and interpreting those charts, contacting analysts, reading their reports, examining the portfolios of her competitors, reading economic forecasts, and possibly even visiting companies.

This appreciation of the value of preparation reveals that when experts solve knowledge-rich problems they engage their environment in far more complex ways than just implementing operators. They interactively probe the world to help define and frame their problems. This suggests that deeper ethnographic studies of everyday problem solving may show a different style of activity than found in formal accounts of problem solving. Theories of knowledge-rich problem solving have become important in the literature since the 1980s. But even these accounts place too little emphasis on the centrality of resources, scaffolds interactivity, and cultural support. There is far more going on in solving a real-world problem than in searching a problem space

PART 3: POSITIVE ACCOUNT - A FEW IDEAS

The view articulated so far is that problem solving is an interactive process in which subjects perceive, change, and create the cues, constraints, affordances, and larger-scale structures in the environment, such as diagrams, forms, scaffolds, and artifact ecologies that they work with as they make their way toward a solution. This looks like the basis for an alternative and positive theory of how people overcome problems in concrete settings. The positive element in situated cognition is this emphasis on agent-environment interaction. All that is missing from such a theory are the details!

How and when do people externalize inner state, modify the environment to generate conjectures, interactively frame their problems, cognize affordances and cues, and allocate control across internal and external resources? These are fundamental questions that must be answered by any positive account that attempts to locate problem solving in the interaction between internal representations or processes and external representations, structures, and processes. If problem solving emerges as a consequence of a tight coupling between inside and outside that is promoted and sustained by culture and the material elements of specific environments, then we need an account of the mechanisms involved in these couplings, and ultimately, of how culture, learning, and the structure of our artifacts figure in shaping that coupling.

It is beyond the goals of this article to present a positive theory, or even a sketch of one. I will discuss, however, four areas in which adherents of situated cognition, in my opinion, ought to be offering theories - areas in which a situationalist might constructively add value to the problem solving literature. My remarks should be seen more as suggestions or desiderata for a situated theory of problem solving than as an actual sketch of one. The four areas are as follows:

1. Hints
2. Affordances
3. Thinking with things
4. Self-cueing

It is important to appreciate that this positive approach is not meant to label other theories - more classical theories - as useless. Insight into problem solving can be found in the gestalt literature, in articles on clinical problem solving, in the study of learning and education, and also in the literature on task environments just criticized. What is offered here should be seen as an addition to those literatures, part of what one day might be a more integrated approach, though the proof that such as theory is viable will require the sort of dedication to ethnography, experimental research, and model building that has so far been lacking in situated cognition.

1. Sketch of a Theory of Hints

Hints come in all forms. They are a part of the natural history of problem solving. Here are a few examples drawn from the classroom. A teacher may tell or show a student how to use a special method or algorithm ("Here's a faster way to determine square root"), give advice on framing the problem ("First state the givens and thing to be shown, then eliminate irrelevant details and distracters"), or suggest useful strategies ("Generate as many lemmas as you can in order to fill your page with potentially useful things"). A fellow student may share illustrations or models, mention an analogous problem, give part of the answer, suggest a way of thinking or representing the problem, and so on. Is there a theory that might explain why a hint is a hint? Hints represent an important element of the culture of problem solving (Kokinov, Hadjiilieva, & Yoveva, 1997).

A simple theory of hints might begin by defining a hint to be a verbal or nonverbal cue that acts like a heuristic bias on search. This would nicely fit the classical theory. Consider the hints people give in the game of I Spy - a child's game in which one player, the spy, gives clues about an object he or she observes and the other player(s) attempt to guess what the object is.

- "You're getting hotter" - metric information - your current guess is better than your last.
- "It's bigger than a loaf of bread" - constraint on candidate generation and an acceptable solution.
- "More over there" — gesture that biases the part of the search space to explore.
- "Try eliminating big categories of things first like 'Is it a living thing?'" - heuristics for efficiently pruning the space.
- "Don't ignore the color" - critical features to attend to that bias generation and evaluation.
- "Back up - abstract away from these sorts of details" - recharacterize the search space.

In thinking of hints in this way, we tie them to their role in both candidate generation and evaluation.

Situated cognition downplays search in a problem space as the determinant of problem-solving behavior. Psychological, social, cultural, and material factors are treated as more important. Nonetheless no approach to problem solving can overlook the importance of both candidate generation and evaluation. Ideas, possibilities, and possible ways of proceeding are always involved in problem solving, as is their evaluation as being fruitful, off the mark, or suggestive. Even in insight problems (Mayer, 1995), where discovering a solution requires breaking functional set - as when a subject suddenly realizes that an object can be used in a nonstandard way - it is still necessary to try out ideas of different ways of using objects. So there is always a component of generating candidate actions and testing their adequacy.

What drives the generation process? It depends on how easy it is to know that one is at a choice point and the set of actions available there. In knowledge-lean problems such as the Tower of Hanoi, there are a small number of possible moves at each step, and an agent knows what these are. The problem is discrete, candidate generation is easy, and state-space search is a good formalization of the problem, even though the agent's psychological task changes when the problem increases in size. In knowledge-lean problems the difficulty lies more in evaluation than in generation - in deciding whether a given option is a good one. Hints therefore ought to offer advice or heuristics for moving in a good direction. For instance, work toward clearing a peg completely; there are no short cuts.

In games like chess where there are many more possible actions at each choice point, a state space still captures the formal structure of the problem, because the agent knows perfecdy well where the choice points are. Given how much knowledge is required for expertise it seems odd to call chess a knowledge-lean problem. In chess it is hard to see the downstream effects of action, because one's opponent is unpredictable. So it is hard to generate plausible chains of "If I do this, then you are likely to do that." The branching factor of the game is large and the space of possible continuations huge. This puts greater pressure on prudent candidate generation at each level of the search space. Accordingly, in chess, hints and suggestions often have to do with biasing generation. For instance, typical rules for opening are these: "Open with a center pawn." "Knights before bishops." "Don't move the same piece twice." "Always play to gain control of the center." Typical rules for middle game include these: "In cramped positions free yourself by exchanging." "Don't bring your king out with your opponent's queen on the board." And in the end game rules include these: "If you are only one pawn ahead, exchange pieces, not pawns." "Don't place your pawns on the same color squares as your bishop."[2]

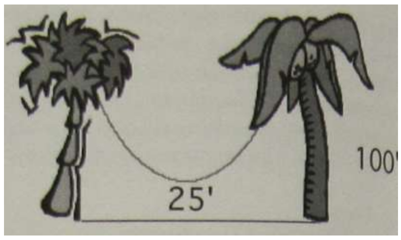


Figure 15.14. Try solving the flagpole problem. Why is it hard? It is not because we cannot think of a way to represent it. We can extract the givens, the solution condition, and constraints. Making an illustration is easy, as shown here. But is this a good illustration? If you have not solved the problem, here is a hint: work in from the extremes to see how the vertical drop changes as the separation between the trees increases or decreases. What kind of hint is this?

In problems that do not count as knowledge-lean, or problems where it is not easy to determine when one is at a choice point and what the options are, the hardest step is often to figure out how to think about the problem, to pose it or frame it in a constructive way. Consider the problem in Figure 15.14. What should one do? Making an illustration is always a good idea, but which illustration, and how should it be annotated? The space of possible drawings and textual actions is huge. Moreover, once a sketch has been made there remains the question of how to proceed.

> Flagpole problem (Figure 15.14): Two palm trees are standing, each 100 feet tall. A 150-foot rope is strung from the top of one of the palm trees to the top of the other and hangs freely between them. The lowest point of the rope is 25 feet above the ground. How far apart are the two flagpoles? (Ornstein & Levine, 1993)

One reason some problems are hard is that people find it difficult to escape their familiar way of proceeding, another instance of mental set. Aspects of a problem may remind them of possible methods or method fragments, but if these seem unpromising, what is to be done? The difficulty hardly seems to lie in finding something to do. It lies in finding an interesting thing to do. Where do ***interesting*** ideas come from? Does staring at a problem help? Should the problem solver do things not directly connected with taking a step forward in the problem, such as doodling, reading more about related problems, talking to colleagues, brainstorming? How many candidate steps come from inside an agent's head and how many come from prompts from artifacts, group dynamics, and so forth?

Early experiments by Gestalt psychologists have shown the value of environmental cues, such as waving a stick or setting a string in motion (Maier, 1970), that call attention to an item that might figure in a solution. But a more theoretically motivated explanation of how these cues trigger candidate generation is needed. And, once subjects do have a new line of thought to pursue, where do they get their metrics on goodness, their ability to discriminate what is an interesting avenue to follow and what looks useless? How does the environment help?

A theory of situated problem solving should explain why hints are successful and the many ways our environments offer us hints on how to solve our problems. At a minimum there is a large body of relevant data to be found in hint-giving and hint-receiving behavior.

2. Affordances and Activity

A second element in a positive theory would explain how people discover candidate steps in a problem solution by interactively engaging their environment. In classical accounts, if a task environment is well defined there is a set of feasible actions specified at each choice point. An agent is assumed to recognize choice points and automatically generate feasible actions. In situations where problems have not been well framed, however, discovering moves to consider can be challenging. In Newell's theory, SOAR (Laird, Newell, & Rosenbloom, 1987), failure to generate a feasible action creates an impasse and a new subproblem to be solved. An alternative suggestion by Greeno and colleagues (Greeno, Smith, & Moore, 1993; Greeno & Middle School Mathematics through Applications Project Group, 1998) is that problem solvers recognize possible moves by being attuned to affordances and constraints. If at first an agent does not see a possible action, she can interact with the environment and increase her chances of discovering it. This is a promising approach that deserves elaboration and study.

The inspiration for seeing problem solving emerging out of interaction with resources and environmental conditions is drawn from Gibson's (1966, 1977, 1979) theory of perception. Gibson regarded an affordance as a dispositional property like being graspable, pullable, or having a structure that can be walked on, sat on, picked up, thrown, climbed, and so on. These properties of objects and environments are what make it possible for an agent with particular abilities to perform actions: pulling X, walking on X, sitting on X, picking up X, throwing X, climbing over X. Agents with different abilities would encounter different affordances. For example, relative to a legless person, no environment, regardless of how flat it is, affords walking. The same applies to other actions and skills. Only relative to an action or activity repertoire does an environment have well-defined affordances.

Because affordances are objective features of an environment, there may be affordances present in a situation that are never perceived at the time. The affordances that are actually perceived depend on the cues available during activity. The more attuned a creature is to its environment, the more it picks up affordance-revealing cues and the more readily it accomplishes its tasks. So runs the theory according to Gibson.

Problem solving, Greeno and colleagues suggest, should likewise be seen as an active, dynamic encounter of possibilities and registering of affordances, constraints, and invariants. When engaged in a task, or when trying to solve a problem, skillful agents recognize aflbrdanees that are relevant to their immediate goals. A cook recognizes the affordances of the stove, the knobs, blenders, and ingredients. Because many of these affordances are representational (e.g., a dial) or rule based (e.g., it is a convention that pulling a lever forward increases rather than decreases the magnitude of whatever it controls), the world of affordances and constraints that a cook is sensitive to must include properties that are socially, culturally, and conventionally constructed. Thus linguistic structures as well as nonlinguistic ones can be affordances for Greeno. In his view, it is not relevant that linguistic and nonlinguistic affordances are learned differendy or that they are grasped differendy: that the way someone knows what depressing a button with the linguistic label "abort" will do is conceptual, whereas the way someone knows that a knife affords hefting or cutting is probably nonconceptual. Both types of possibilities for action are "seen" and qualify as affordances for Greeno.

This inclusiveness represents a major extension of the concept of affordance and constraints, as understood in ecological psychology, but if it can be made to work, it permits seeing problem solving to be the outcome of a more embodied encounter with cultural resources. Greeno and colleagues assume that people can perceive or register affordances for activities that are quite complex. For instance, they assume that the more familiar we are with tools, such as hammers, screwdrivers, chisels, knives, machine saws, and lawn mowers, the more we can see opportunities for using them. Carpenters can make hundreds of perceptual inferences about wood. And someone who mows his lawn every week can see when grass is ready for its next cutting or too dense for a given lawn mower. This is an important and useful extension but not without difficulties.

Because using tools invariably involves mastering practices, the affordances that tool users perceive must be complex. Imagine the affordances, constraints, and invariants that a chef must pick up when preparing an egg sunny-side down. First a hard edge must be found for cracking the egg cleanly, then the egg itself must be held correctly when opened, the frying process must be monitored, and the egg shifted at the right time so that it does not stick. Flip, ping has its own complexities. Throughout the process a cook must be sensitive to the preconditions of actions, the indicators that things are going well or beginning to go awry, and the moment-by-moment dynamics of cooking. A sunny-side-down cooking trajectory, under normal conditions, is supposed to be an invariant for a competent cook. In the theory of attunement to affordances and constraints, all these cues, affordances, constraints, and invariants of normal practice are precisely the things that skilled agents are supposed to be attuned to.

In extending affordances to include the affordances that situations provide for tools (in the hands of competent users), Greeno and colleagues pushed the concept well beyond what can be perceived through normal perception - even ecological perception. Their theory goes even further, though, to include affordances for reasoning. Such things as marking up diagrams, working with illustrations, and manipulating symbolic representations are all actions that experienced agents can perform and that serve as steps in reasoning.

For instance, when a student of geometry sees an illustration as in Figure 15.15, it is assumed that she can recognize a host of affordances for construction. To someone who has learned to make triangular constructions it is natural to look at Figure 15.15b and imagine dropping perpendiculars, as in Figure 15.15c. With practice and a little prodding, most students realize that the area of Figure 15.15b is the same as the rectangle in Figure 15.15a, and both are base times height (b x h). The proof involves noting that the triangles in Figure 15.15c are congruent. If someone cut out the left-hand triangle, he could paste it on the right and produce a rectangle just like Figure 15.15a. It is possible to prod students to recognize and generalize

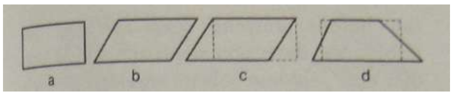


Figure 15.15. For a student of geometry, 15.15b affords many different types of constructions, including dropping perpendiculars from vertices as in 15.15c, adding diagonals, bisecting, and so forth. These affordances for constructions can be projected mentally onto the figure or added by pencil or pen on the figure itself. Although there are an infinite number of constructions the figure affords, a geometer only considers those constructions that are part of the standard practices. For instance, in 15.15CI a clever student might realize that the area of a trapezoid can be proved similar to the area of a rectangle by noticing that triangles constructed through the midsection of each vertical side are congruent and create a rectangle whose length is (top + bottom)/2.

this idea by giving them paper and scissors and the chance to cut and paste. When they perform this physical action a few times, or perform the construction on paper, they come to see an invariance preserved under a shearing transformation. Clever students who are able to prove the equality of area theorem more analytically may even be able to recognize opportunities for making less obvious constructions, as in Figure 15.15c!, thereby generalizing further the equality of area theorem and recognizing the invariance under further transforms.

The idea that an experienced subject can become attuned to both affordances and relevant invariants is powerful but constitutes a theory of problem solving only if it offers predictions. This is a challenge for an attunement theory. It is not a lost cause but is certainly one that has yet to be met. Exactly when will a subject pick up affordances for construction and, given the vast number of such affordances, exactly which ones will she attend to? This last concern applies to all theories of affordances but is especially problematic for those that presume affordances for reasoning.

To see how serious this challenge is, consider Figure 15.16. The two shapes in Figure 15.16a are topologically invariant. There is a stretching operation that transforms Figure 15.16a into Figure 15.16b. How many people see it? Because there are an infinite number of ways to stretch the figure, it requires an insightful mind to envisage the transform. Figure 15.16b gives one set of deformations that shows the equivalence. Yet how many people can see even the first transform from 1 to 2? A theory of affordance pickup must tell us who, when, and why some people can see this invariance and who, when, and why certain others cannot. It also must explain why the most useful affordances are the ones that come to mind.

Education researchers would love to provide a theory of attunement learning, though so far advances have been empirical rather than theoretical. For instance, Sayeki, Ueno, and Nagasaka (1991) found that students who were given a chance to play with a deck of cards as shown in Figure 15.17a, and then to chat about their activity, were soon able to recognize the invariance of the area in the figures shown in Figure 15.17b. Physical experience with the deck helped to imbue students with a strong sense of the transformations that preserve area. Presumably subjects who played with clay structures shaped into the structure shown in Figure 15.16a would similarly have an easier time following the constructions in Figure 15.16b.

Greeno and colleagues viewed this consequence of practice as support for their theory that problem solving is the result of attunement to affordances, constraints, and

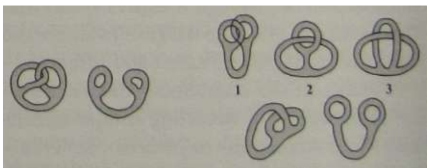
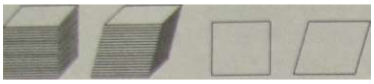


Figure 15.16. All the figures shown in 15.16a and 15.16b are topologically the same: 15.16b is a sequence of transforms that is meant to prove the equivalence of figures in 15.16a. As with most proofs, not everyone is able to "see" the validity of each inferential step.



a b

Figure 15.17. Students were shown a deck of cards as in 15.17a. The front face initially formed a square. As the deck was pushed around it became apparent that many four-sided and non-four-sided shapes could be constructed. The attribute common across all transformations is that the area of the front side always remains constant. The equivalence of the two images in 15.17b became obvious, intuitive, known in an embodied way.

in variance. Practice and engagement leads to improved attunement, which in turn leads to better solution exploration.

This is a rather different view than the classical account for two reasons. First, in the classical account, problem solving always involves search in a problem space. In the attunement account, problem solving involves evaluating perceived or registered affordances. This bypasses the need for a problem space where feasible actions are internally represented. Choice points are encountered rather than internally generated. And the choice points encountered depend on the actions actually taken by the agent - including mental projection actions. This means that the environment an agent encounters is partly co-constructed by its actions. As an agent begins to see things differently, especially ways it can project or construct, it partly creates new task affordances. Constructions support new projections and so on, though it must be said that there remains a need to triage or evaluate the many affordances registered and projected.

Second, in the attunement theory, transfer is the result of detecting similar affordances and constraints or invariances. Transfer is a direct consequence of being able to detect affordances and constraints, whether or not the agent has much grasp of the structure of a problem. Because similar affordances can be found in many situations, including those that pose very different problems, agents are more likely to try things out before grasping deep analogies. fa affordance detection approach would prediet that problem solvers would be actively engaging their environment, both by projecting and by acting. In the classical theory, by contrast, the reason a subject is able to transfer problem-solving expertise from one problem to another is that the subject is able to detect deep structural similarities, and so methods found successful in the source domain are mapped over to the target domain. There is no deep need to interact with the problem environment.

If the theory of attunement were better specified, these advantages would be substantial. But the theory, as developed to date, falls short of details. How, for instance does a subject choose which affordance to act on? There are infinitely many affordances available in any situation. This is particularly true once the notion of an affordance has been broadened to include the possibility of projecting structure or creating structure. In Figure 15.16a, for instance, there is no end of ways to deform the clay model. The challenge is to know which of the many ways to pursue.

In Greeno et al. (1998) it is suggested that dynamical system models may help here. It is easy to see why. If problem solving is a dynamic encounter with an affordance-rich environment, the language of attractors and repellors ought to be helpful in explaining why some affordances are perceived rather than others, or why some actions are pursued rather than others. Ecological psychologists have often advanced dynamical system accounts. But again, prior to a more complete account of the details, we can only wonder what controls this dynamic encounter. Discourse about basins of attraction and repulsion - the favorite terms in discussions of dynamical systems - sound remarkably like the discourse of gradient ascent: an action is selected if it yields a higher value. This is the method of hill climbing and is one of the cornerstone methods of problem-space search. Yet it is not enough. To explain much of the observed behavior of subjects on knowledge-lean problems, it has been found

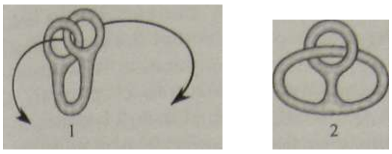


Figure 15.18. By adding annotations or other markings it is easier to envision the way this structure can be deformed. Such cues help to anchor projection and facilitate envisionment. It is a rich area for study.

necessary to introduce other weak methods such as difference reduction, depth-first search, breadth-first search, minimax search, abstraction, and others. Attunement theory owes us the details of how these sort of mechanisms of control can be implemented. If on the other hand dynamical systems are going to supplant the need for these additional control mechanisms, then we need to know more about how they will cope with the details of control that have seemed so useful in the classical theory.

There are further questions. Why are affordances and constraints sometimes visible and sometimes not? In Figure 15.16b, for example, it is hard for most people to see the legality of the transform between 1 and 2. But in Figure 15.18, with the simple addition of annotations, or hints, or material anchors, it is much easier. The details of an attunement model should explain the biases on affordance and constraint detection. It should tell us the types of detection errors that subjects make and why. It should tell us the biases we observe in human problem solving. And it should tell us the time course of affordance-constraint detection: why some are quickly seen and others are detected slowly. There is substantial opportunity for theory and experiment here. It may eventually pay off but it is clear there is much to be done.

3. Thinking with Things

How people use artifacts, resources, and tools as things to think with delineates a third class of phenomena that we would expect a positive theory of situated problem solving to study and explain. In discussing how milkmen used their bottle cases to calculate efficient plans, we considered how overlearned patterns supported or embedded in artifacts - such as the regular pattern that forty-two bottles makes in a forty-eight-bottle milk case - could be used to help them solve specific planning and accounting problems. Experienced algebraicists use patterns in matrix representations to recognize properties of linear equations, and experienced milkmen use patterns of bottles, both present and imagined. These patterns figure in special-case solutions they have learned. The infamous weight watcher's example shows a similar trick: the physical form and size of cottage cheese when dumped from its carton can be used to support highly specific arithmetic operations that would be harder for most subjects to perform in their head. The idea of cutting a regular cylinder in half is so intuitive that for many people it is understood more directly than fractions. They can think with the parts the way they can think with symbols. C. S. Peirce (1931-1958) first mentioned this idea - that people use external objects to think with - in the late nineteenth century, when he said that chemists think as much with their test tubes as with pen and paper.

The notion that we use things to think with, that we distribute our thinking across internal and external representations or manipulables, is relatively uncontentious when the artifacts used are symbolic, such as written sentences, illustrations, numbers, or even gestures. Much ethnographic research has shown in detail how people use artifacts to encode information: how people represent information in external structures and then manipulate those external structures and read off the results. The same applies to gesture. People use body gestures to help them think in context, both when they think individually and when they think as a group or in a team. Gestures help them remember ideas, formulate thoughts, and not just supplement vocalization. A slightly different idea attaches to the use of artistic media. No one would deny that a sculptor is thinking or solving certain problems when using clay any more than someone would deny that an author is thinking or solving problems with the help of pen, paper, and written sentence. The medium has important properties that the artist or author is trying to exploit. Interaction is essential to the process. And though one might say that all the relevant cognition occurs inside the head of the human, the physical materials, scaffolds, tools, and structures are an important factor in the outcome. They help to structure the affordance landscape.

The question of how people think with things, how the determinants and dynamics of cognition depend on properties of artifacts and the context of action, is unquestionably at the heart of the theories of situated, distributed, and interactive cognition. But as with other tenets, it is in need of greater elaboration and empirical study. To date, the majority of studies have been confined to ethnographic examinations of particular cases. This is a necessary step given the importance of details. But little attention has been paid to the distinction between using objects to solve special-case problems - a method that reflects memorized techniques - and using objects or systems in more general ways as an intrinsic part of reasoning and problem solving. The distinction is important because if most instances of thinking with things are highly specific, if most are cases where dedicated tools are used to solve domain-specific problems, then too much of the rest of problem solving is left out and any hope of a more general theory of problem solving looks bleak. How much did we really leam about problem solving by observing weight watchers partition cottage cheese?

The theory situated cognition owes us is one that will explain how people harness physical objects to help them reason and solve problems. What characteristics must a thing to think with have if it is to be effective, easy to use, and handily learned?

Although a comprehensive theory would provide a principled taxonomy of the ways we can think with things, a tiny step toward this theory can be taken by looking at how people co-opt things to perform computations. Computation is about harnessing states, structures, or processes to generate rational outcomes. It is about using things to find answers. The most familiar forms of computation are digital, the manipulation of symbol strings, as in math, engineering, or computer science. But all sorts of nonsymbolic systems can be harnessed to perform computations. For instance a slide rule is an analog mechanism for performing multiplication, addition, and a host of other mathematical operations. It does not have the precision of a calculator, or the range of functions of many other digital devices. But because it preserves key relationships - linear distance along each scale of the rule is proportional to the logarithm of the numbers marked on it - moving the slides in the correct way allows a user to perform multiplications. It can be used to simulate the outcome of digital multiplication because it preserves key relationships. It replaces symbol manipulation with manipulation of physical parts.

The same can be said, though less easily, for illustrations, sketches, and three-dimensional models. For instance, one of the most common ways we think with things is by using them as model fragments. A structure or process can be said to model or partly model another if it is easy to manipulate and examine the model and then read off implications about the target structure or process. An extreme example is seen in the architectural practice of building miniature three-dimensional models of buildings. The operations that architects perform on their small models are sufficiently similar to actions that inhabitants will perform on or in full-sized buildings that architects can try out on the model ways the full-sized building may be used. They can act on the model instead of the real object. This saves time, effort, and cost because mistakes have few consequences in models and simulations. This can even be done using two-dimensional diagrams, as shown by Murphy (2004), who recorded how architects reason about the uses of a building by bringing their bodies into interaction with their architectural drawings.

The formal part of a thinking-with-things theory should provide the analytic tools for evaluating why certain things can function successfully as things that can be thought with. Much of this work has already been done in other fields. For example, in math it is well understood that a powerful technique of reasoning is to map structures into different representational systems. Problems that may be hard to solve in one system, say, Euclidean geometry, may be easy to solve in analytic geometry. This mapping between representational systems may not seem to be a computation, but it is. It is a trick that is widely used. For instance, when sailors plot a course, they typically use more than one map. Every map represents the world under a projection that preserves some relations while distorting others. Mercator projections are good for plotting course but bad for estimating distance or area. Sailors overcome this problem by plotting their course in maps of different projection. They convert information from one map to another. The result is that they are able to track their location, distance, bearing, and speed more accurately and more quickly than they otherwise could (Hutchins, 1995). They think across maps. The result of using multiple maps is like using a fulcrum: it allows a weak user to do some heavy lifting because the maps or relation between the maps does much of the work.

An even more complicated way of using things to think with is found when people use the very thing they are interested in as its own model. For example, people solving tangrams seem to use the tangram pieces as things to think with. So do people when they assemble things without first reading the assembly manual. When Rod Brooks (1991a, 1991b) initially offered the expression "use the world as its own model," he meant that a robot would be better off sensing and reacting to the world itself than by simulating the effects of actions in an internal model of the world prior to selecting action. If a robot were sufficiently tuned to the regularities of its environment and the tasks it needed to perform, it could be driven by a control system that would guarantee with high probability that the robot would eventually reach its goals, though not necessarily by the shortest path. If this sounds similar to the theory of attunement to affordances and constraints it is because it is the same theory though restricted here to problems such as trajectory planning, grasping, and physical manipulation, and without the requirement of mental projection.

Although such an approach may seem the antithesis of problem solving, it is a legitimate way of dealing with real-world problems - but only if certain conditions are met. Specifically:

- The world must be relatively benign. Moving toward goals can only rarely lead to disastrous downstream consequences (Kirsh, 1991,1996).
- The world must be reversible. If you do not like your action, you can undo it and either return the world to the condition it was in before, or you can find a new path to any of the states you could reach before.

When can these conditions be counted on? Assembly tasks, math tasks, and puzzles, such as tangrams, support undoing without penalty. The Tower of Hanoi is another classic task supporting reversibility. And checkers, chess, and other competitive games usually have a form - correspondence chess, checkers - where subjects can search over possible moves directly on the board before deciding how to move. In other tasks, even if one cannot reverse the action, subjects can get a second chance to solve the problem. So as long as it takes no longer to try out an action in the world than trying it out in a model, there are advantages to acting directly in the world. First, subjects gain precision in the representation of the outcome because nothing is more precise than the real thing. Second, subjects gain practice in bringing things about, which can be of value where skill is required. Third, subjects save having to formulate a plan in their head, try it out in a model, and then execute the plan in the world. The execution phase is folded into the discovery phase.

But there are many other tasks where actions are not reversible and search in the world is a bad idea. In cooking, for example, as with many tasks, after an initial preparation phase when a recipe is selected and ingredients gathered, there is an execution phase where there is no turning back. There are many places where a plan can be modified or updated to deal with errors, setbacks, and unexpected outcomes. But in most tasks there are commitment points where it is inappropriate to do anything but the next move.

Most tractable analyses of thinking approach thought as a computational activity whether that computation is said to occur inside the head on internal representations, outside the head through the use of physical objects, or in the interaction between inside and outside. An even more profound approach to thought sees it as a mechanism for extending our perception, action, and regulation. This is a radical vision of how cognition is shaped through our interaction with artifacts.

The core idea in this more *enactive* theory of thought is that thinking is somehow tied up with the way we encounter and engage the world. A violinist encounters the world through his violin in a way that depends essentially on the violin. Violin problems are encountered only in violin playing and constituted in the interaction of player and instrument. The same applies to people who bicycle, whitde, manipulate cranes, or solve higher-math problems. The material instruments, representational languages, and sensorimotor extensions that artifacts provide offer new modes of experience and involvement with the world. The problems that arise are somehow essentially tied to those interactions and therefore cannot be properly analyzed until we understand what those who have those skills experience as problematic.

I personally think this is an exciting area of research. As with other areas this one lis much in need of clarification and empirical exploration. Moreover, there is a danger that this activity-centric model makes thought so relativistic and hermeneutic that only violinists can understand other violinists only sculptors can understand other sculp' tors, and so on. Although I believe such concerns can be answered, they highlight that there are philosophical and methodological problems as well as empirical ones that such an approach may face.

4. Self-Cueing and Metacognitive Control of Discovery

What controls search? In the classical theory the firing of productions, a form of associative memory, drives search. If a rule exists whose preconditions match patterns currently in working memory, then it is triggered, and unless other rules match current conditions and are therefore also triggered, that solitary rule fires and causes a change in working memory. Owing to this focus on what is in current working memory there is an inescapable *data-driven* bias to search, both on candidate generation and evaluation. Only a change in working memory can trigger new productions. Environmental state enters working memory through perception. A positive interactionist or situationist theory ought to provide an account of how we interact with our environments to overcome the lock that data-driven thinking has on our creativity. It must provide a theory of the dynamics driving the interactive exploration of a problem during the search for solution. One special line of inquiry looks at the way we self-cue, how we alter the cue structure of the environment to stimulate new ideas or candidate generation.

A nice example of how as data-driven creatures we self-cue to improve performance can be found in the way people play Scrabble (see Figure **15.19**). The basic problem in Scrabble is to find the highest-value words that can be made from the letters

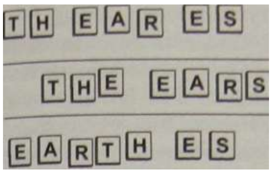


Figure 15.19. When people play Scrabble, they scan pieces on their tray, look for openings on the board, and search through a developing space of possible word and word fragments. Most people also periodically move the letters on their tray. Sometimes this is to hold their current candidate. But often it is to highlight high-frequency two- and three-letter combinations that figure in words.

on one's tray after placing them in a permissible position on the board. The challenge for a psychological theory is to explain how the external cues, comprised of the ordered letters on tray and board, mutually interact with a subject's internal lexical system. The challenge for an interactionist account is to show how subjects intentionally manage the arrangement of letters on their tray to improve performance.

In Maglio, Matlock, Raphaely, Chernicky, and Kirsh (1999), an experiment was performed to test the hypothesis that people who rearrange letters externally self-cue and consequently perform better than those who do not. The task was similar to Scrabble in that subjects were supposed to generate possible words from seven Scrabble letters. Unlike Scrabble, though, they just called out words as they recognized them; there were no constraints coming from the board or from values associated with letters. In the hands condition, participants could use their hands to manipulate the letters; in the no hands condition they could not. Results showed that more words were generated in the hands condition than in the no hands condition, and that moving tiles was more helpful in finding words that were more distant from the opening letter sequence.

Evidently, self-cueing helps beat the data-driven nature of cognition. We conjectured that the reason moving tiles is helpful is that when subjects rearrange tiles they are in effect jumping to a new place in their internal lexical space. By lexical space we meant a system of discrete nodes consisting of letter sequences. How close one node is to another in this space is determined by such things as phonemic closeness *(bear* is close to *bare* and *air),* graphemic closeness *(bear* is close to *ear),* and perhaps others things such as semantic closeness *(bear* is close to *lair).* In our simulation we defined closeness as graphemic closeness alone. Thus two nodes are neighbors in or simplified model of internal lexical space if they can be reached by shifting or removing a few letters or adding a letter if there is an additional letter on the tray. Hence, *bear* is close to *bar, bare, bra* and if there is another *e* on the tray, then the word *beer* is also a graphemic neighbor. Letter sequences that do not form words, such as *bre ebra,* are also lexemic neighbors but less strongly attractive if they are infrequent sequences in English.

In discussing our findings we reasoned that although subjects can in principle apply as many mental transforms in lexemic space as they want and so, in both the no hands and hands condition, they can in principle move arbitrarily far from the letter layout on their physical tray. We expected that in practice subjects would tend to get stuck in the lexical neighborhood near to the groupings on their tray. This idea reiterates the data-driven nature of Scrabble.

The power of using one's hands comes from the difference in the constraints on physical movement and *mental movement.* A just-so story runs like this. The letters that subjects see exert a pull on their imagination much like an elastic band. The farther that subjects go in their lexemic space from the sequence on the tray the stronger the tension pulling them back. This tension is purely internal. In physical space, there is nothing preventing players from moving their tiles ever further from their first layout. It is easy to break the elastic-band effect. This means that a subject can jump to a new place in lexical space, and so activate a new basin of neighbors just by rearranging his or her tiles. The new arrangement will prime or cue a new set of lexical elements and make it more likely that new words will be discovered.

This approach to problem solving, if intentional, looks a lot like metacognition. It is reminiscent of the behavior of Ulysses (Elster, 1979), who recognized that he could not overcome the lure of the Sirens except by binding himself. To achieve his goal, given that he knew he would act inappropriately in the situation, he altered the situation. We do the same when we move our eyes. Given that our visual system is automatic and data driven, we cannot help but see what we look at. But we still can avert our eyes or direct them to other things. The decision to look elsewhere is often intentional, and when it is, it counts as metacognitive if it is based on reflectively exploiting the way our visual system works. The same applies to moving tiles in Scrabble and a host of problem-solving strategies we rely on in other domains. In math, for instance, we often copy onto a single page the lemmas we generated over many pages. This increases the chance of seeing patterns. In algebra, students soon learn that it helps to be neat because it is easier to see relations and patterns. Indeed, the strategy of rearranging the environment to stimulate new ideas when candidate generation slows down is pervasive. The principle relied on is self-cueing and metacognitive control. (For a more complete discussion of the role of metacognition in reasoning and learning, see Kirsh, 3004.3

The moral for research in problem solving should be clear: by studying more carefully how cues and affordance landscapes bias cognition, new interactive strategies will be discovered that show unanticipated ways subjects use the environment to shape their problem-solving cognition.

Final Discussion and Conclusion

By exposing how cognition is closely coupled to its social, material, and cultural context the situationalist approach has called attention to the deficiencies of the classical theory of problem solving. It has forced us to reconsider the form an adequate theory should take. Is a general theory of problem solving possible? Is there enough resemblance between the actions that problem solvers take when solving or overcoming problems to hope to discover a general theory of the dynamics of problem solving, regardless of domain? I argued that such a theory must at least tell us for a given problem and situation how much of the control is to be found in internal processes, directed by such internal resources as problem-space representations and heuristic search, how much of the control is to be found in the setup or design of the environment influencing cognition through the affordances, cues, and constraints that a culture of activity has built up over time, and how much is to be found in the dynamic interaction between internal and external resources.

This is a tall order. It requires developing a set of supporting theories that explain how cues, constraints, and affordances affect how a subject thinks and acts. This point came up repeatedly: first when discussing the processes of registration, then when discussing interactivity and epistemic actions, scaffolds, and resources, and later when I discussed the direction a positive theory might take: explaining attunement to symbolic affordances and cues, explaining how subjects respond to hints, how they self-cue, and how they think with material things. A general theory of problem solving would have to pull all these supporting theories together to explain how subjects move back and forth among cues, scaffolds, visible attributes and the mental projections, structures, and problem spaces those subjects maintain on the inside. It would be an interactionist theory.

The situationalist approach, secondly, has exposed what is wrong with studying problem solving in constrained laboratory contexts, disconnected from the settings in which those mental and interactive processes originally developed. Subjects adapt to the world they live in. The internal costs of problem solving depend substantially on how experienced a subject is, whether the problem is presented to him or her in a familiar way, and how effectively he or she can exploit the surrounding resources, cues, affordances, and constraints. This concern with cognitive costs and benefits is consistent with the general theme of model building in cognitive science. But putting it into practice means paying closer attention to the details of natural contexts. This cannot be done without close ethnographic and microanalytic study.

As a negative theory, situated cognition has been a success. But if approaches are judged by their positive theories, situated cognition has been a failure. All efforts at creating a substantive theory of problem solving have been underspecified or fragmentary. And it is too early to know whether the next dominant theory will bear a situationalist stamp.

Building on the critique presented in Part One and Part Two, I posed a set of questions and initial approaches that might indicate the direction situated research should pursue next. These include an analysis of hints and scaffolds, symbolic affordances and mental projection, thinking with things, self-cueing and metacognition, and an enactive theory of thought. Some of these studies are being undertaken outside of cognitive and computational psychology. They explore how people interactively populate situations with extra resources and how they exploit those resources to simplify reasoning. Some of these studies, however, are not yet being undertaken. The bottom line, it seems to me, is that it is not enough that we recognize the central insight of situated cognition - that the environment provides organization for cognitive activity, that the world enables and supports such activities 1 we must go further. We must explain how internal control processes work with these supports and organizational structures to regulate intelligent activity.

Acknowledgments

I am happy to thank the editors for their valuable suggestions and moral support during the writing of this chapter and for comments by Rick Afterman and Jim Greeno. Joachim Lyon carefully read the entire manuscript and gave thoughtful remarks throughout. Support for this work came in part from ONR Award # N00014-02-3-0457.

Notes

1 In computational complexity theory, problems are ranked according to the resources needed to solve their hardest instance, as measured by the number of steps and the amount of memory or scratch paper the best algorithm will use. For instance, solving a traveling salesman problem with two cities is trivial because there is only one combination of cities, one tour, to consider (A, B). But as the number of cities increases, die number of candidate tours that have to be written down and measured increases exponentially (as there are as many possible tours as sequences of all cities). It is the shape of this resource growth curve that determines the complexity class of a problem. A problem is said to be in the class of NP-complete if, in the worst case, the best algorithm would have to do the equivalent of checking every tour, and the test to determine whether one tour is better than another is itself not an NP-complete algorithm. There are infinitely many problems that are harder than NP-complete ones, and infinitely many problems that are easier. The easier ones have polynomial complexity.

2 Taken from "The Thirty Rules of Chess." Retrieved May 31, 2008, from http://www.chessdryad.com/education/sageadvice/thirty/index.htm.

References

Agre, P., & Chapman, D. (1987). Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence* (pp. 268-272). Menlo Park, CA: AAAI Press.

Baker, B. S. (1994). Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM,* 41(1), 153-180.

Behrend, D. A., Rosengren, K., & Perlmutter, M. (1989). Parent presence. *International*

Journal of Behavioral Development, 12(3), 305-

Brooks, R. A. (1991a). Intelligence without reason. In *Proceedings of the* 1991 *Joint Conference on Artificial Intelligence* (pp. 569-595). San Francisco, CA: Morgan Kaufmann.

Brooks, R. A. (1991b). Intelligence without representation. *Artificial Intelligence,* 47,141—159.

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18,* 32-42.

Carraher, T. N., Carraher, D. W., & Schliemann, A. D. (1985). Mathematics in the streets and in the schools. *British Journal of Developmental Psychology,* 3, 21-29.

Chambers, D., & Reisberg, D. (1985). Can mental images be ambiguous? *Journal of Experimental Psychology: Human Perception and Performance,* 11, 317-328.

Clark, A. (1997). *Being there: Putting brain, body, and world together again.* Cambridge, MA: MIT Press.

Devlin, A. S., & Bernstein, J. (1996). Interactive wayfinding: Use of cues by men and women. *Journal of Environmental Psychology, 16,* 23-39-

Elster, J. (1979). *Ulysses and the sirens.* Cambridge: Cambridge University Press.

Gibson, J. J. (1966). *The senses considered as perceptual systems.* Boston, MA: Houghton Mifflin.

Gibson, J. J. (1977). The theory of affordances. In R. E. Shaw & J. Bransford (Eds.), *Perceiving, acting, and knowing {pp.* 67-82). Hillsdale, NJ: Lawrence Erlbaum.

Gibson, J. J. (1979). *The ecological approach to visual perception.* Boston: Houghton Mifflin.

Gick, M., & Holyoak, K. (1980). Analogical problem solving. *Cognitive Psychology, 12,* 306-355.

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

Greeno, J. G., & Middle School Mathematics through Applications Project Group (1998). The situativity of knowing, learning, and research. *American Psychologist,* 53(1), 5-26.

Greeno, J. G., & Moore, J. L. (1993). Situativity and symbols: Response to Vera and Simon. *Cognitive Science,* 17,49-60.

Greeno, J. G., Smith, D. R., & Moore, J. L. (1993). Transfer of situated learning. In D. K. Detterman & R. J. Sternberg (Eds.), *Transfer on trial: Intelligence, cognition, and instruction* (pp. 99-167). Norwood, NJ: Ablex.

Hamberger, J. (1979). *Music and cognitive research: Where do our questions come from? Where do our answers go?* (Working Paper Ser No 2.). Cambridge: Massachusetts Institute of Technology, Division for Study and Research in Education.

Hutchins, E. (1995). *Cognition in the wild.* Cambridge, MA: MIT Press.

Hutchins, E., & Palen, L. (1997). Constructing meaning from space, gesture, and speech. In L. B. Resnick, R. Saljo, C. Pontecorvo, & B. Bürge (Eds.), *Discourse, tools, and reasoning: Essays on situated cognition* (pp. 23-40). Heidelberg, Germany: Springer-Verlag.

Karp, R. M. (1986). Combinatorics, complexity, and randomness. *Communications of the ACM* 29(2), 98-109.

Kershaw, T. C. (2004). Key actions in insight problems: Further evidence for the importance of non-dot turns in the nine-dot problem. In K. Forbus, D. Gentner, & T. Regier (Eds.), *Proceedings of the Twenty-Sixth Annual Conference of the Cognitive Science Society* (pp. 678-683). Mahwah, NJ: Lawrence Erlbaum.

Kirsh, D. (1991). When is information explicitly represented? In P. Hanson (Ed.), *Vancouver studies in cognitive science* (Vol. 1, pp. 340-365). New York: Oxford University Press.

Kirsh, D. (1995a). Complementary strategies: Why we use our hands when we think. In J. D. Moore & J. F. Lehman (Eds.), *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society* (pp. 212-217). Hillsdale, NJ: Lawrence Erlbaum.

Kirsh, D. (1995b). The intelligent use of space, *Artificial Intelligence,* 72(1-2), 31-68.

Kirsh, D. (1996). Today the earwig, tomorrow man? In M. Boden (Ed.), *Artificial life* (pp. 237-261). Oxford: Oxford University Press, (Reprinted from *Artificial Intelligence,* 47(3), 161-184.)

Kirsh, D. (2003). Implicit and explicit representation. In L. Nadel (Ed.), *Encyclopedia of cognitive science* (Vol. 2, pp. 478-481). London: Nature Publishing Group.

Kirsh, D. (2004). Metacognition, distributed cognition and visual design. In P. Gärdenfors & P. Johansson (Eds.), *Cognition, education and communication technology* (pp. 147-179). Hillsdale, NJ: Lawrence Erlbaum.

Kirsh, D., & Maglio, P. (1995). On distinguishing epistemic from pragmatic actions. *Cognitive Science, 18,* 513-549.

Kokinov, B., Hadjiilieva, K., & Yoveva, M. (1997). Is a hint always useful? In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Lawrence Erlbaum.

Kotovsky, K., & Fallside, D. (1989). Representation and transfer in problem solving. In D. Klar & K. Kotovsky (Eds.), *Complex information processing* (pp. 69-108). Hillsdale, NJ: Lawrence Erlbaum.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987) SOAR: An architecture for general intelligence. *Artificial Intelligence,* 33, 1-64.

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science, 11,* 65-99.

Lave, J. (1988). *Cognition in practice: Mind, mathematics and culture in everyday life.* Cambridge: Cambridge University Press.

Lave, J. (1996). The practice of learning. In S. Chailkin 8c J. Lave (Eds.), *Understanding practice: Perspectives on activity and context* (pp. 3-32). New York: Cambridge University Press.

Lesgold, A. M. (1988). Problem solving. In R. J. Sternberg 8c E. E. Smith (Eds.), *The psychology of human thought* (pp. 188-213). New York: Cambridge University Press.

Luchins, A. S. (1942). Mechanization in problem solving: The effect of Einstellung. *Psychological Monographs,* 54,1-95.

Luchins, A. S., 8c Luchins, E. H. (1970). *Wertheimer's seminars revisited: Problem solving and thinking* (Vols. 1-3). Albany: State University of New York Press.

Maglio, P., Madock, T., Raphaely, D., Chernicky, B., 8c Kirsh, D. (1999). Interactive skill in Scrabble. In *Proceedings of Twenty-First Annual Conference of the Cognitive Science Society.* (pp. 326-330). Mahwah, NJ: Lawrence Erlbaum.

Maier, N. R. F. (1930). Reasoning in humans: I. On direction. *Journal of Comparative Psychol-OQT,* 10,115-143.

Maier, N. R. F. (1970). *Problem solving and creativity.* Belmont, CA: Brooks/Cole.

Mayer, R. E. (1995). The search for insight. In R. J. Sternberg 8c J. E. Davidson (Eds.), *The nature of insight* (pp. 3-32). New York: Cambridge University Press.

Miller, G. A., Galanter, E., 8c Pribram, K. H. (1960). *Plans and the structure of behavior.* New York: Holt, Rinehart 8c Winston.

Murphy, K. M. (2004). Imagination as joint activity: The case of architectural interaction. *Mind, Culture and Activity,* 11(4), 270-281.

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

Newell, A., 8c Simon, H. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice Hall.

Norman, D. A. (1988). *The psychology of everyday thingX.* New York: Basic Books.

Ornstein, A. C., 8c Levine, D. U. (1993). *Foundations of education* (5th ed.). Boston: Houghton Mifflin.

Paige, J. M., 8c Simon, H. (1979). Cognitive process in solving algebra word problems. In H. A. Simon (Ed.), *Models of thought.* New Haven, CT: Yale University Press.

Papadimitriou, C., 8c Yannakakis, M. (1988). Optimization, approximation, and complexity classes. In *Proceeding? of the Twentieth Annual ACM Symposium on Theory of Computing* (pp. 229-234). New York: ACM Press.

Peirce, C. S. (1931-1958). *The collected papers of Charles Sanders Peirce* (C. Hartshorne, P. Weiss [Vols. 1-6], 8c A. Burks [Vols. 7-8], Eds.). Cambridge, MA: Harvard University Press.

Pretz, J. E., Naples, A. J., 8c Sternberg, R. J. (2003). Recognizing, defining, and representing problems. In J. E. Davidson 8c R. J. Sternberg (Eds.), *The psychology of problem solving* (pp. 3-30). Cambridge: Cambridge University Press.

Reitman, W. R. (1964). Heuristic decision procedures, open constraints, and the structure of ill-defined problems. In M. W. Shelly & G. L. Bryan (Eds.), *Human judgments and optimality* (pp. 282-315). New York: John Wiley.

Rogoff, B., & Lave, J. (Eds.) (1984). *Everyday cognition: Its development in social context.* Cambridge, MA: Harvard University Press.

Sayeki, Y., Ueno, N., 8c Nagasaka, T. (1991). Mediation as a generative model for obtaining an area. *Learning and Instruction, 1,* 229-*M²- -M*

Scribner, S. (1984). Studying working intelligence. In B. Rogoff 8c J. Lave (Eds.), *Everyday cognition: Its development in social context* (pp. 9-40). Cambridge, MA: Harvard University Press.

Scribner, S. (1986). Thinking in action: Some characteristics of practical thought. In R. J. Sternberg 8c R. K. Wagner (Eds.), *Practical intelligence: Nature and origins of competence* in *tkr everyday world* (pp. 13-30). Cambridge: Cambridge University Press.

Simon, H. A. (1969). *The sciences of the artificial.* Cambridge, MA: MIT Press,

Simon, H. A. (1979). *Models of thought* (Vol. 1). New Haven, CT: Yale University Press.

Simon, H. A., & Hayes, J. (1976). The understanding process: Process isomorphs. *Cognitive Psychology, 8,*165-190.

Sternberg, R. J. (1997)- *Thinking styles.* New York Cambridge University Press.

Sternberg, R. J., & Davidson, J. E. (1983). insj h(in the gifted. *Educational Psychologist,* 18,51_ 57-

VanLehn, K. (1989). Problem solving and cog nitive skill acquisition. In M. I. Posner (Ed) *Foundations of cognitive science* (pp. 527-;^)' Cambridge, MA: MIT Press.