An Extended Theory of Human Problem Solving

Pat Langley (langley@csli.stanford.edu)
Seth Rogers (srogers@csli.stanford.edu)
Computational Learning Laboratory
Center for the Study of Language and Information
Stanford University, Stanford, CA 94305 USA

Abstract

Human problem solving has long been a central topic in cognitive science. We review the established theory in this area and note some phenomena it does not address. In response, we present an extended framework, cast within a theory of the cognitive architecture, that provides one account of these phenomena and that we illustrate using the familiar Tower of Hanoi puzzle. We conclude by discussing other extensions to the standard theory and directions for future research.

Introductory Remarks

Research on human problem solving has a venerable history that played a central role in the creation of cognitive science. Early studies of problem-solving behavior on puzzles and other novel tasks led to many insights about representation, performance, and learning that now underpin the field. Moreover, computational models developed from these studies remain some of the most detailed and precise accounts of human cognition.

However, the past decade has seen considerably less attention paid to this important topic, presumably because many researchers believe that it is sufficiently well understood. In contrast, we maintain that the standard theory of problem solving, although basically accurate, is still incomplete, and we need additional work, at the level of both theoretical principles and specific computational models, to extend our understanding of this complex cognitive activity.

In this paper, we review traditional accounts of problem solving and note some important omissions that require further effort. After this, we present an extended framework that we have embedded in Icarus, a computational theory of the human cognitive architecture. We then illustrate our points using a classic task, the Tower of Hanoi. In closing, we discuss other variations on the traditional theory and topics for future research.

The Standard Problem Solving Theory

The standard theory of problem solving, initially outlined by Newell, Shaw, and Simon (1958), focuses on how humans respond when they are confronted with unfamiliar tasks. Early work focused on abstract problems like proving theorems in propositional logic and solving the Tower of Hanoi puzzle. Later research adapted the framework to explain cognition in semantically rich domains like solving word problems in physics and thermodynamics, as well as addressing the observed behavioral differences between domain experts and novices.

The theory makes a number of claims about human cognition. The most basic is that problem solving involves the mental inspection and manipulation of list structures. Newell and Simon (1976) later refined this into their *physical symbol system* hypothesis, which states that symbolic processing is a necessary and sufficient condition for intelligent behavior. Another central claim, termed the *problem space* hypothesis, is that problem solving involves search through a space of candidate states generated by operators.

A more detailed aspect of the theory is that, in many cases, problem solvers utilize means-ends analysis (Newell & Simon, 1972). This class of search methods involves a combination of selecting differences between the desired and current states, selecting operators that will reduce the chosen differences, and either applying the operators or creating subproblems to transform the current states into ones in which they can apply. This requires one to chain backward from aspects of the goal state to find relevant operators and determine useful subgoals. However, with experience this novice strategy is replaced in experts with forward chaining that leads directly to the goal (Larkin et al., 1980).

Nevertheless, closer analyses of human behavior on novel tasks have suggested that this story is incomplete and that the actual situation is more complicated. Here we make some additional observations that are not addressed by the standard theory.

- *Problem solving occurs in a physical context.* Puzzles like the Tower of Hanoi are typically presented in some physical form, with solutions relying on manual actions and tests for legal moves requiring visual inspection. This physical setting simplifies the task by providing an external memory, but it also introduces irrelevant features.
- *Problem solving abstracts away from physical details, yet must return to them to implement the solution.* For instance, when solving the Tower of Hanoi, humans appear to search through an abstract problem space that describes states in terms of disk-peg configurations and operators as transitions between them. They ignore the details of grasping and moving required to demonstrate the solution, but they can execute these actions when necessary.

- *Problem solving is seldom a purely mental activity, but rather interleaves reasoning with execution.* In the Tower of Hanoi, the problem solver may reason backward to select an intended move, but typically makes that move as soon as it is legal, without waiting to make sure that he can solve the remainder of the problem from that point.
- *Eager execution of partial plans can lead the problem solver into physical dead ends that require restarting the task.* For the Tower of Hanoi, initially moving the smallest disk to the goal peg on an even-disk problem means one cannot solve the puzzle without later retracting this objective. However, once the problem solver has made such an execution error, he is unlikely to repeat it on later attempts.
- *Learning from successful solutions transforms backward chaining search into informed skill execution.* When a person first solves the Tower of Hanoi, he resorts to means-ends analysis, but sufficient experience on the task replaces this with an automatized execution procedure that involves no search.

Naturally, because these facets of human problem solving have not received attention as interesting phenomena, computational models have also tended to ignore them. In the next section, we present an expanded theory of problem solving that begins to remedy these oversights.

We should clarify that do not we view our framework as inconsistent with the standard theory, but rather as augmenting it. Neither are we the first to raise the need for such extensions, as we note below when we discuss each issue at more length. However, we are the first to address them in a consistent manner within a theory of the human cognitive architecture. Moreover, we note that the two best-known frameworks – Soar (Laird, Rosenbloom, & Newell, 1986) and ACT-R (Anderson, 1993) – have been augmented to address some of these issues, but we view these changes as retrofits rather than deeper revisions at the architectural level.

An Extended Problem Solving Theory

We are now ready to present an extended theory of human problem solving that moves beyond the abstract framework provided by the standard version. This forms a key part of Icarus, an account of the human cognitive architecture. We will present only those aspects of Icarus relevant to problem solving, but we have presented more complete descriptions elsewhere (Choi et al., 2004). We will use the phenomena listed above to organize our points.

Physical Setting of Problem Solving

First, Icarus is an architecture for physical, embodied agents. Although the framework incorporates many ideas from traditional work on cognitive modeling, it maintains that cognition is closely tied to perception and action, and one cannot build an Icarus model that is not linked to some environment external to the agent. For the Tower of Hanoi, we have developed a simulated world that contains disks, pegs, and a hand. Each object has attributes that describe its height, width, and position, and the hand also has a status of open or closed.

Like other cognitive architectures, Icarus operates in cycles, but processing involves more than cognitive activity. On each cycle, the agent perceives objects in its immediate environment and their descriptions are deposited into a short-lived perceptual buffer. The system may also carry out actions associated with primitive skills; for the Tower of Hanoi, these correspond to grasping and ungrasping a disk, as well as moving the hand up, down, left, or right. Naturally, we could model both perception and action at a finer granularity, but this level is sufficient to make our point about the embodied nature of problem solving.

Of course, the situated cognition movement has also argued that human behavior occurs in a physical setting, with Zhang and Norman (1994) even reanalyzing behavior on the classic Tower of Hanoi puzzle in these terms. Newell and Simon (1972, pp. 800–803) were quite aware that much of their subjects' behavior occurred in this context; they simply chose not to distinguish between internal and external states in their models. However, we have developed a cognitive architecture that treats the physical side of problem solving as a central tenet, as contrasted with frameworks like Soar and ACT-R, which view embodiment as peripheral to cognitive processing.

Abstraction from Physical Details

As noted, although humans have the perceptual-motor abilities necessary to execute solutions to tasks like the Tower of Hanoi, their mental problem solving occurs at a more abstract level. Icarus models this capability in two ways. One involves a long-term conceptual memory that, on each cycle, recognizes instances of generic situations and adds them to a short-term memory that contains current beliefs about the agent's environment. For instance, our model for the Tower of Hanoi includes concepts for noting when a peg is empty and when a disk is the smallest on a peg.

The other involves a long-term skill memory that describes higher-level skills, like moving a disk, in terms of subskills like grasping a disk, lifting it off a peg, carrying it from one peg to another, lowering the disk onto a peg, and ungrasping it. On each cycle, the architecture selects for execution a path through the skill hierarchy with application conditions that match. These skills are durative, in that they may take many cycles to complete; for example, this holds for carrying a disk from one peg to another in our Hanoi environment. We assume the execution of such complex skills occurs in an automatized way that demands very few attentional resources.[1]

Thus, Icarus' sensory-motor control cycle provides both abstract descriptions of the environment and ways to execute abstract operators. This lets much of problem solving occur at an abstract level, which verbal protocols consistently suggest is how humans deal with such tasks. However, our framework is more constraining than the physical symbol system hypothesis, in that

[1]Because Icarus' hierarchical skills are interpreted reactively, they are much less rigid than traditional hierarchies.

problem states are always grounded in real or imagined physical states, and problem-space operators always expand to primitive skills with executable actions. We believe a more accurate claim is that cognition relies on a *symbolic physical system*, which comes closer to Johnson-Laird's (1989) views on thinking with mental models.

Interleaved Cognition and Execution

In addition to modules for conceptual inference and skill execution, ICARUS includes a module for means-ends analysis. As in traditional accounts, we assume the problem solver begins with a declarative goal that describes aspects of the desired state. This must be stated as an instance of some known concept, so that the agent cannot even formulate a problem unless it has a long-term structure that can recognize its solution. This assumption is novel, but not a difference we will focus on here.

Skills contain not only application conditions and subskills, but also descriptions of their expected effects. If the agent has a skill in long-term memory that would achieve the goal, it checks to see whether that skill can be applied in the current setting. If so, then the agent simply executes the stored solution in the environment. But if it lacks such a skill, as it would for an unfamiliar task, it resorts to means-ends problem solving.

In this case, two situations can arise. The agent may know a skill that would achieve the goal concept, but it cannot yet be applied. Here the module adds the instantiated precondition of the skill, which is stated as a single concept, to a goal stack, then repeats the process by looking for skills that will achieve this subgoal. Alternatively, the agent may have no skill that achieves the goal concept, in which case it examines the concept definition and adds one of its unsatisfied subconcepts to the goal stack, making it the focus of problem solving.

This process follows the standard theory fairly closely, but it diverges when the agent retrieves a skill that can be applied in the current state. Unlike traditional models, ICARUS does not apply the skill mentally and continue reasoning from the new internal state. Instead, it executes the skill physically in the environment, interrupting cognitive activity for as many cycles as needed, then resumes internal problem solving in the context of the new situation. This may lead to more execution, if the precondition of the next skill on the stack is met, or to more subgoaling, if more remains to be achieved.

As a result, ICARUS never retains in goal memory a complete plan about how it will solve the problem, but only a single path backward through its concept and skill definitions that would lead toward its initial goal. The advantage of this scheme is that it places a much lighter memory load on the problem solver by using the environment as an external store.[2] Again, this account does not conflict with Newell and Simon, who never claimed that problem solving took place entirely in the human mind, but it does contrast with most follow-on efforts.

[2]We should note that humans can handle simple problems mentally, so a compromise model like that reported by Gunzelmann and Anderson (2003) may offer a better account.

Problem Restarts

Of course, the disadvantage of this mixed strategy is that the problem solver may make an incorrect choice, such as which subgoal it should achieve first. Heuristics can reduce the chances of this happening, but they can seldom eliminate it entirely. Traditional systems, including the General Problem Solver (Newell & Simon, 1972), responded by backtracking in the problem space. This produces a systematic strategy of depth-first search that does not match the details of human problem solving.

ICARUS engages in a limited form of depth-first search, but only if it can accomplish this by pushing subgoals onto, or popping them from, the current goal stack.[3] Once it has made a physical move, the system will not consider retracing this step as part of a systematic search. Moreover, the goal stack retains information about subgoals the system has accomplished in pursuit of higher-level goals, and it does not consider skills that would undo them. Thus, the agent may find itself in situations where it has achieved some subgoals necessary for a solution but where it has no actions available.

In such cases, the problem solver gives up on its current attempt and starts the problem again. Anzai and Simon (1979) note two instances of this strategy in their protocols, and many readers will recall taking such steps on complex or tricky tasks. Restarting on a problem requires physically moving it to the initial configuration, which can be viewed as a form of backtracking, but it is certainly not the systematic version usually discussed in the literature. ICARUS assumes that this restarting process plays a key role in human problem solving.

We should note one earlier model, reported by Jones and Langley (1995), combined restarts with means-ends analysis. However, their EUREKA system carried out purely mental problem solving and had limited access to memories of previous solution attempts, which meant it could make the same errors in choice repeatedly. Humans may retrace some errorful steps, but they also try to avoid these choices on later attempts. To model this tendency, ICARUS stores the goal stack associated with a move later judged to be mistaken and uses it to eliminate the move from consideration on successive trials. One might view this as systematic search across problem attempts, but it still differs from most models of problem solving and, we hold, comes closer to human strategies.

Learning from Problem Solutions

ICARUS' storage of long-term structures about what moves to avoid constitutes a simple form of learning, but the traces are highly specific and meant only to avoid previous errors. However, people can also acquire more general knowledge about how to solve a class of problems, and the architecture incorporates a mechanism with this capability. This learning method is interleaved with performance, it operates in an incremental manner, and it is cumulative in that it builds on knowledge acquired earlier, all characteristics of human behavior.

[3]We are also exploring memory-limited goal stacks that are more consistent with recent studies of problem solving.

The learning process is driven by impasses that occur during execution and problem solving, which in turn lead to pushing entries onto the goal stack. Because ICARUS aims to avoid such impasses in the future, the architecture creates a new skill whenever it pops the stack upon achieving its current goal. However, recall that the means-ends module creates two distinct types of subgoal that require slightly different forms of learning.

One type occurs when the agent selects a skill instance $S2$ to achieve a goal G, but, upon finding its start condition unsatisfied, selects another skill instance $S1$ so as to achieve it. When the agent has executed both skills successfully and reached the goal, ICARUS constructs a new skill that has the same name and arguments as the goal concept G, the two skills $S1$ and $S2$ as ordered subskills, and a start condition that is the same as that for $S1$. In addition, arguments and other peceived objects are replaced by variables in a consistent manner.

The other form of learning occurs when the agent finds no skill to achieve a goal G, and thus creates subgoals for the unsatisfied conditions of G's concept definition. If the agent achieves each subgoal in turn, the architecture constructs a new skill that has the same name and arguments as the goal concept G, the subgoals that are achieved as ordered subskills, and a start condition based on the subconcepts of G that held initially. Both mechanisms name new skills after the goals they achieve, which leads naturally to disjunctive and recursive definitions.

After learning, when the agent encounters a problem that is isomorphic to one it has solved before, it avoids an impasse by accessing the stored skill. The architecture instantiates this skill and executes it, continuing until achieving the goal that led to its retrieval. Similar events transpire if the system encounters any subproblem it has solved previously, which can produce transfer to new tasks with familiar subtasks. Note that the agent transitions from the backward-chaining search of means-ends analysis to forward execution. This is consistent with observations of novice-expert shifts (Larkin et al., 1980), but clarifies that the "forward chaining" occurs in the environment, rather than in the agent's mind.

A Model for the Tower of Hanoi

To illustrate these ideas further, we will consider an ICARUS model for the Tower of Hanoi puzzle. The system begins with some 14 concepts that describe configurations of disks and pegs, organized in a hierarchy that has five levels. We also provide the system with five primitive skills for manipulating disks, along with four higher-level skills that correspond to problem-space operators,[4] some of which appear in Table 1. In addition, we embed the system in a simulated environment with three disks and three pegs, which it perceives on each cycle and can alter using the actions in its primitive skills.

The agent begins by perceiving three disks stacked on the leftmost peg, along with two other empty pegs, and inferring a number of higher-level concepts from these percepts. These do not satisfy its goal of having a tower composed of the same disks on the rightmost peg, so it attempts to retrieve a skill that will transform its current state into the desired one. Failing to retrieve such a skill, it decomposes the tower goal into three subgoals, each of which involves getting a separate disk on the rightmost peg, and selects among them.

Because the model knows it can move the smallest disk to the goal immediately, it selects this alternative, although the choice causes problems later. The agent selects a skill that will achieve this subgoal and, since its conditions are satisfied, executes it in the environment. This skill invokes subskills for grasping, lifting, carrying, lowering, and ungrasping the disk, most of which take multiple cycles to complete. When it has finished the move, it pops the subgoal from the goal stack.

[4] The system requires four operators because it must distinguish between cases in which the moved disk is/is not alone on the 'from' peg and between similar cases for the 'to' peg.

Table 1: Some ICARUS skills for the Tower of Hanoi.

```
(pickup-putdown (?disk ?from ?to)
 :start    ((pickup-putdownable ?disk ?from ?to))
 :ordered  ((grasp-disk ?disk)
            (lift-disk ?disk)
            (carry-disk ?disk ?from ?to)
            (lower-disk ?disk ?to)
            (ungrasp-disk ?disk))
 :effects  ((no-disk-on ?from)
            (on-peg ?disk ?to)
            (only-disk-on ?disk ?to)))
(unstack-putdown (?d1 ?d2 ?from ?to)
 :start    ((unstack-putdownable ?d1 ?d2 ?from ?to))
 :ordered  ((grasp-disk ?d1)
            (lift-disk ?d1)
            (carry-disk ?d1 ?from ?to)
            (lower-disk ?d1 ?to)
            (ungrasp-disk ?d1))
 :effects  ((on-peg ?d1 ?to)
            (top-disk-on ?d2 ?from)
            (only-disk-on ?d1 ?to)))
(lift-disk (?disk)
 :start    ((on-peg ?disk ?from)
            (no-smaller-disk-on ?disk ?from)
            (grasped ?disk))
 :actions  ((*vertical-move ?disk 1))
 :effects  ((above-peg ?disk ?from)))
(carry-disk (?disk ?from ?to)
 :percepts ((peg ?from xpos ?fxpos)
            (peg ?to xpos ?txpos))
 :start    ((above-peg ?disk ?from)
            (grasped ?disk))
 :actions  ((*horizontal-move ?disk ?fxpos ?txpos))
 :effects  ((above-peg ?disk ?to)))
(lower-disk (?disk ?to)
 :start    ((above-peg ?disk ?to)
            (no-disk-on ?to)
            (grasped ?disk))
 :actions  ((*vertical-move ?disk -1))
 :effects  ((on-base ?disk ?to)))
(lower-disk (?disk1 ?to)
 :start    ((above-peg ?disk1 ?to)
            (on-peg ?disk2 ?to)
            (no-smaller-disk-on ?disk1 ?to)
            (no-smaller-disk-on ?disk2 ?to)
            (grasped ?disk1))
 :actions  ((*vertical-move ?disk1 -1))
 :effects  ((on-disk ?disk1 ?disk2)))
```

The system next selects the subgoal of getting the second smallest disk on the goal peg, along with the skill instance that would move it there. However, the start condition for this skill is not matched, so the model pushes a subgoal onto the stack to achieve it. But it cannot find any skill that would accomplish it and not undo the subgoal regarding the smallest disk it has already achieved.

As a result, it pops the goal stack and marks the proposed skill as failed, then selects another that it hopes will move the second smallest disk to the rightmost peg. However, it finds that this skill leads to the same problem and abandons it as well. Thus, the system pops the subgoal for the second smallest disk and replaces it with the subgoal of moving the largest disk to the goal peg. But this encounters the same problems as before, which eventually causes it to abandon this subgoal.

Lacking any further options, the model realizes its first subgoal was an error, so it stores the context in which it made this selection in order to avoid the subgoal in future attempts. The agent then resets the environment to the initial configuration and tries the puzzle again. This time it avoids the previous mistake but instead attempts to get the second smallest disk onto the goal peg, which leads to similar problems. After some effort, it decides to abandon this attempt and restart the problem again.

The model continues in this fashion, chaining backward to select subgoals and skills and executing the latter whenever they are applicable. On its eighth attempt, the agent selects the subgoal involving the largest disk first, followed by the second largest disk, which together let it solve the problem successfully. Along the way, the system creates a new skill whenever it achieves a subgoal that requires executing two or more component skills.

For the three-disk puzzle, the model acquires 13 skills that build on initial hierarchy. When given the same problem, the agent retrieves the relevant learned skill and executes it without invoking means-ends analysis. There are two ways to instantiate this skill, one that moves disks to the goal peg and another that moves them to the other peg. The model makes the right choice half the time, in which case it makes the correct sequence of moves in a reactive manner. When it makes the wrong choice, the model fails and restarts, either to repeat its error or make the right one on the next round. Humans often make similar errors even after practice on the task.

We have included this example to illustrate the ways in which ICARUS' behavior is qualitatively consistent with the phenomena discussed earlier. Other researchers have presented models of behavior on the Tower of Hanoi, including some (e.g., Anderson, 1993) that match closely the timing observed on the task. Such detailed modeling has benefits, but it should not be expected in the early stages of developing a unified theory that covers new phenomena, when qualitative evaluation is appropriate.

Related Theoretical Extensions

We are hardly the first to propose extensions to the standard theory of problem solving. For example, ICARUS' learning mechanisms are closely related to the chunking process in SOAR (Laird et al., 1986) and to knowledge compilation in ACT-R (Anderson, 1993). All three create new structures that reduce an agent's efforts on future tasks. A key difference is that the older architectures determine conditions on learned rules by analyzing dependencies in reasoning traces, whereas ICARUS simply stores generalized elements from its goal stack. Moreover, their learning mechanisms eliminate intermediate structures to reduce internal processing, whereas ICARUS' mechanisms operate in a cumulative manner that leads to the creation of hierarchical structures.

We have also mentioned Jones and Langley's (1995) EUREKA, which incorporated a modified means-ends problem solver that combined search down a single path with restarts upon failure. Their model did not learn either generalized chunks or hierarchical skills, but it did include a form of analogical search control based on problem-solving traces accessed through spreading activation. ICARUS stores instances of specific failures, but uses them in a more limited way to keep from repeating mistakes. EUREKA also utilized a flexible version of means-ends analysis which preferred operators that reduced more differences and fell back on forward-chaining search when no other choices were available. We intend to incorporate this idea into future versions of ICARUS.

Jones and Langley also used their model to account for insight in problem solving, a topic which Ohlsson (1992) has discussed at more length. In his treatment, problem solving sometimes leads to impasses that cannot be overcome in the initial problem space. Insight occurs when the problem solver restructures the space in some manner that makes a solution possible. Ohlsson suggested three distinct restructuring mechanisms that can produce this effect. In some ways, his proposal constitutes a more radical extension to the standard theory than our own, but it also expands on Newell et al.'s original analysis without rejecting its basic contributions.

Other researchers have extended the standard framework to explain the benefits of diagrams in problem solving. For example, Larkin and Simon (1987) have argued that, besides serving as external memories, diagrams reduce search by grouping elements that are used together, utilize location to group information about a given element, and support perceptual inferences that are easy for humans. This extension seems closely related to our concern with the embodied context of problem solving, although diagrams have a somewhat different character from the physical structures in many puzzles and games.

Concluding Remarks

Although our extended theory of problem solving is roughly consistent with the phenomena we presented earlier, our framework would clearly benefit from further evaluation and refinement. There remain many questions about how much humans abstract away from the physical details of the problem setting, how often they return to this setting during their search, how tightly they integrate cognition with execution, exactly when they restart on difficult problems, and how rapidly learning lets them acquire automatized strategies. ICARUS makes specific predictions about these issues, but testing them will require carefully studying traces of problem-solving behavior with them in mind.

We also know the current theory remains incomplete in some ways. For instance, when humans first encounter a novel problem, they appear to spend time observing and manipulating the objects involved, presumably to construct an abstract representation of the task. Icarus does not model this early learning process, although Gobet and Simon's (2001) work on the role of discrimination learning in game playing suggests one promising avenue. Moreover, people seem to form generalizations not only about what steps to take in solving problems, but about what steps to avoid. Icarus stores specific memories to avoid repeating errors, but, unlike Soar and Prodigy, it does not form generic rejection knowledge.

Modeling these aspects of learning is important, but we should also remedy some drawbacks of our performance framework. Although Icarus cannot backtrack over execution errors, it does carry out depth-first search within each problem-solving attempt, whereas humans seem far less systematic. Future versions should incorporate the memory-limited strategy used by Eureka, which seems closely related to the progressive deepening scheme observed in chess players. Finally, although means-ends analysis has been widely implicated in problem solving, it is not the only method that humans use to solve problems. Forward-chaining search appears to dominate on complex tasks like chess in which backward chaining from the goal is impractical. We should add this ability to Icarus and specify when this form of search is invoked and how it interacts with means-ends analysis.

In summary, our extensions to the standard theory have added some important items to our store of knowledge about human problem solving. These include the claims that problem solving typically occurs in a physical context yet abstracts away from physical details, that problem solving interleaves mental reasoning with execution, that this interaction can lead to dead ends which require problem restarts, and that learning transforms backward chaining into informed execution. Moreover, we have embedded these ideas in Icarus, a cognitive architecture that we have used to develop models for behavior on the Tower of Hanoi and other domains.

However, we should remember that the scientific process itself involves problem solving. The extensions we have reported here are simply further steps toward a full understanding of human cognition. These must be followed by the heuristic application of additional operators, and possibly even by backtracking, before we achieve this worthwhile but challenging goal.

Acknowledgements

This research was funded in part by Grant HR0011-04-1-0008 from DARPA IPTO and by Grant IIS-0335353 from the National Science Foundation. Discussions with Glenn Iba, David Nicholas, Dan Shapiro, and others contributed to many of the ideas presented here.

References

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

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

Choi, D., Kaufman, M., Langley, P., Nejati, N., & Shapiro, D. (2004). An architecture for persistent reactive behavior. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems* (pp. 988–995). New York: ACM Press.

Gobet, F. & Simon, H. A. (2001). Human learning in game playing. In M. Kubat & J. Furnkranz (Eds.), *Machines that learn to play games*. Huntington, NY: Nova Science.

Gunzelmann, G., & Anderson, J. R. (2003). Problem solving: Increased planning with practice. *Cognitive Systems Research*, *4*, 57–76.

Johnson-Laird, P. N. (1989). Mental models. In M. I. Posner (Ed.), *Foundations of cognitive science*. Cambridge, MA: MIT Press.

Jones, R., & Langley, P. (1995). Retrieval and learning in analogical problem solving. *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society* (pp. 466–471). Pittsburgh, PA: 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., McDermott, J., Simon, D. P., & Simon, H. A. (1980). Expert and novice performance in solving physics problems. *Science*, *208*, 1335–1342.

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, *4*, 317–345.

Minton, S., Carbonell, J. G., Knoblock, C. A., Kuokka, D., Etzioni, O., & Gil, Y. (1989). Explanation-based learning: A problem solving perspective. *Artificial Intelligence*, *40*, 63–118.

Newell, A., Shaw, J. C., & Simon, H. A. (1958). Elements of a theory of human problem solving. *Psychological Review*, *65*, 151–166.

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

Newell, A., & Simon, H. A. (1976) Computer science as empirical enquiry: Symbols and search. *Communications of the ACM*, *19*, 113–126.

Ohlsson, S. (1992). Information processing explanations of insight and related phenomena. In M. Keane & K. Gilhooly (Eds.), *Advances in the psychology of thinking* (Vol. 1). London: Harvester-Wheatsheaf.

Zhang, J., & Norman, D. A. (1994). Representations of distributed cognitive tasks. *Cognitive Science*, *18*, 87–122.