Integrating, Not Debating, Situated Action and Computational Models: Taking the Environment Seriously

Michael D. Byrne
School of Psychology
Georgia Institute of Technology
Atlanta, Georgia 30332-0170
`byrne@cc.gatech.edu`

Abstract

A recent issue of the journal *Cognitive Science* (1993, vol. 17, no. 1) centered around a debate between two "camps" within the field, the "situated action" (or SA) camp and the "traditional," symbol processing camp. Though the debate in that journal suggests that, at some levels, symbol processing and SA are incommensurable, this paper disputes that view. If the message of the SA community is taken to be that traditional approaches neglect the importance of the environment, then not only is the message an important one, but the typical symbol processing system is guilty as charged. However, this does not mean that, in principle, symbol processing systems must have this limitation. The two approaches can work hand-in-hand to produce more general and more accurate computational models. A framework of building models of the environment and having models of cognitive agents work with those models is proposed, from which a smooth integration of SA and symbol processing is not only possible, but desirable. The framework proposed here is instantiated with a production system called S-CAPS, and the efficacy of building models of both the problem-solver and the problem environment is demonstrated.

Introduction

A recent issue of the journal *Cognitive Science* (1993, vol. 17, no. 1) centered around a debate between two "camps" within the field, the "situated action" (or SA) camp and the "traditional," symbol processing camp. While an engaging debate, this discussion shed more heat than light on the subject—both sides presented arguments for their points, with neither side yielding enough ground to reach a point of productive dialogue. The central aim of this paper is to make the case that these positions are not only not incommensurable, but complementary. First, the positions of both camps (admitted caricatures) will be outlined, then a general approach to integration will be laid out, and finally, an example of the integration will be provided.

The Situated Position

At the most extreme, the situated position is quite close to Skinner's position of environmentalism: all of the interesting behavior of an organism can be understood by examining the environment in which the behavior occurs. Though many of the SA proponents claim that their position is not, in fact, behaviorist, this might well be attributed to a lack of appreciation for Skinner's work. Take, for example the following illustration from Skinner (1966):

> The differences between rule-following and contingency-shaped behavior is obvious when instances are pretty clearly only one or the other. The behavior of a baseball outfielder catching a fly ball bears certain resemblances to the behavior of of the commander of a ship taking part in the recovery of a reentering satellite. Both move about on a surface in a direction and with a speed designed to bring them, if possible, near a falling object at the moment it reaches the surface. Both respond to recent stimulation from the position, direction, and speed of the object, and they both take into account the effects of gravity and friction. The behavior of the baseball player, however, has been almost entirely shaped by contingencies of reinforcement, whereas the commander is simply obeying rules derived from the available information and from analogous situations.

The parallel between this illustration and the example of Truckese navigators vs. European navigators found in the preface to Suchman's (1987) *Plans and Situated Actions* is striking. (Interestingly, nowhere in Suchman's book or any of the SA replies to Vera and Simon (1993) is Skinner referenced.) The position taken by many of the SA advocates is similar to that of Brunswik (1956), more recently discussed by Hammond (1986), that current work in psychology and decision-making lacks "representative design;" that the environments typically studied by behavioral scientists are not relevant to the environments in which the behavior typically occurs. Again, this is clearly parallel to the arguments put forth by another prominent member of the SA community in reference to classroom education's failure to generalize to supermarket situations. (Lave, 1988).

One might take the position, then, that what the field of cognitive science ought to do in regard to SA is the same thing that it did with Skinner's behaviorism—reject it lock, stock, and barrel. If this is the obvious response, why has SA won over many followers? One hypotheses about the source of the SA community's success is that the cognitive "revolution" threw the proverbial baby out with the bath water. In rejecting behaviorism, the cognition community went to far, and threw out the environment as well. By ignoring the influence of the environment on behavior, key features of the interaction between humans and the world are missed, such as the rich, non-deterministic nature of complex behavior. Another often-commented upon aspect of real environments is that they are dynamic systems that constantly change, whether the agent is acting upon them or not (e.g. Kirlik, Miller, & Jagacinski, 1993).While perhaps taking certain points a to an extreme, the SA community has a legitimate and important message.

The Symbol Processing Position

Traditional cognitive science has been largely concerned with formalism, both in the sense of structural formalism in theories (e.g. computer simulations) and formal empirical methodology (i.e. controlled experimentation). This has led cognitive science to be concerned mainly with easily-formalizable domains. These are typically "high-level" cognitive activities, such as reading and playing chess. While an understanding of reading and chess is certainly important, it can be argued that the proportion of a person's life spent engaged in such high-level activity is small.

The focus on formal, high-level tasks has, however, had its impact on symbolic models. Models created by researchers in this area often are essentially unable to interact dynamically with the world. The standard form of interaction with the world is that a model will receive some kind of input, take that input and go off computing (not again looking outside itself) until it is finished processing, and then "act" upon the world by producing some output. Models of the world, on the other hand, are usually impoverished and static if they are attempted at all.

Take as an example a typical production system such as OPS5. While strict OPS5 systems are not commonly used in cognitive science, there are several systems based or similar to OPS5: Soar (Newell, 1990), CAPS (Just & Carpenter, 1992), and others. Note that the commentary here applies to most symbol processing systems, not just production systems.

OPS5 systems rarely act upon the real world, or even a model of the real world. OPS5 has two memory systems: working memory and production memory. Pattern-matching takes place, and those productions that match patterns present in working memory are instantiated. In OPS5, only one of the instantiated productions is fired, meaning the actions associated with the production are executed. The actions of productions do one of three things: delete items in working memory, add items to working memory, or change the attributes of items in working memory. Note the commonality: all of these actions affect the system's working memory. There is nothing external to the system! (See Figure 1 for a depiction of this process.)

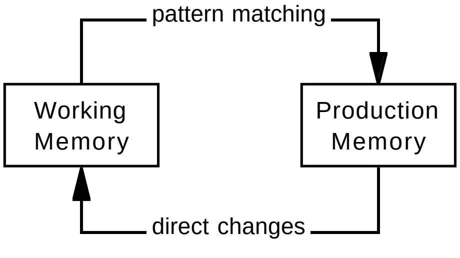

Figure 1. OPS5-style system

"Actions" taken by the system are typically represented by adding working memory elements that represent the state of the world after the action (postconditions) and deleting those elements that are no longer true. There is nothing that represents the action itself, and system actions are the only processes that modify the contents of working memory. Actions are completely deterministic in the sense that the results of an "action" are always the same. In addition, these "actions" are always successful. There is no dynamic world, as only the system can modify working memory. The state of both the agent and the world is totally controlled by the knowledge present in the agent—a sort of telekinesis—and completely represented in a memory system that resides within the agent. Not surprisingly, such systems normally have an unbounded space in which to represent the world, which is another strange and unrealistic property that lends itself to the belief in large "map" and "plan" type structures.

Thus, while Vera and Simon (1993) claim that it is possible to make computational models "situated," this is generally left as exactly that: a possibility. With few exceptions, the symbol processing approach has paid little more than lip service to modeling the environment.

Integrating the Environment

One of the central problems with the label "situated action" is that it implies that some action is "situated" and some action is not. In Lave's (1988) example of grocery store arithmetic, for example, calculation in the market is characterized as "situated" while calculation in the classroom is not).[1] Surely, though, both the classroom and the grocery store are environments, and both of them can be formally modeled. And cognitive scientists are still actively modeling students doing mathematics. Why not, then, maintain models of both the cognitive agent and of the environment in which it is behaving? A truly integrated framework must not only *be able* to model both, but *must* model both. In order for a model built upon such a framework to run at all, explicit models of both the agent and the environment should be required, thus addressing the interaction of the agent and its situation.

Constructing such models will obviously require modifications to the way traditional symbolic models are constructed, but these modifications need not be all that severe, as implicitly suggested by Suchman (1987, p. 63): "We walk into a situation, identify its features, and match our actions to it." This is remarkably close to the way a traditional production system of if-then rules works, so such a system will once again be taken as a starting point, though many of these changes could conceivably be applied to other symbolic systems as well.

What are the properties that a framework for integration should support? One of these features is that the framework should distinguish between internal and external

---

[1] Some SA advocates may, in response, claim that all action is situated—if so, the label SA is misleading. This is one of the primary reasons some SA advocates have taken to the term "situativity" (Greeno, personal communication). Either way, it does not affect the point made here.

representations. That is, the framework should make it clear what things are “stored” in the environment, and which are represented internally by the agent. In such a framework, the impact of such “distributed cognition” should be clear. People obviously perform the same task (such as multiplying ten-digit numbers) better in some environments (with paper and pencil) than in others (without any external aid). There needs to be a performance difference implied by a difference in locus of representation.

Second, the agent should be reactive. That is, the agent’s behavior should not be strictly tied its goals and/or plans. In one current model, ACT-R (Anderson, 1993), productions can fire only if they can match to the current goal of the agent. Thus, if the agent is multiplying two numbers and a fire breaks out in the room, the agent is doomed to perish in the flames since actions like “exit the room” are irrelevant to the current goal context. Goal management needs to be looser and the agent more reactive than in most traditional models.

Another distinction that should be enforced by the framework is a distinction between internal and external actions taken by the agent. The processes by which an agent model acts upon the world and by which an agent model acts upon internal representations should be different. Since internal representations are “owned” by the agent, direct action on those is appropriate. Also, the agent should not control the outcome of taking action. This is the purpose of the external model; it is the part of the model that should make explicit what can and cannot be done.

The environment model should be responsible for managing the contents of the system’s external memory. This includes not only handling changes of state in response to the agent’s actions, but changes over time due to the dynamics of the system. In a model of an air traffic controller, for instance, the environment model must handle both the controller’s actions and the actions taken by the planes the controller is monitoring.

It should be noted that there have recently been systems proposed with some of these features. The most noteworthy of these systems include Teton (VahLehn & Ball, 1991), which has loose goal management, DiBS (Larkin, 1989),)which has a split between internal and external memory and the Phoenix project (Cohen, Greenberg, Hart, & Howe, 1989), which contains an environmental model. These are useful beginnings, but do not completely address all of the issues raised above.

A First Attempt at Integration

This section will describe a framework that is intended to meet the criteria listed in section 2. This framework is based on CAPS (Just & Carpenter, 1992), and will be referred to as S-CAPS (for “situated CAPS”). The goal of S-CAPS is to move away from traditional “internal only” agent models typical of symbolic models by adding an environment model, as well as specifying how the environment model and the agent model interact. The basic flow of information and control in S-CAPS is presented in Figure 2. Note that rectangles designate structures internal to the agent, and ovals external structures.

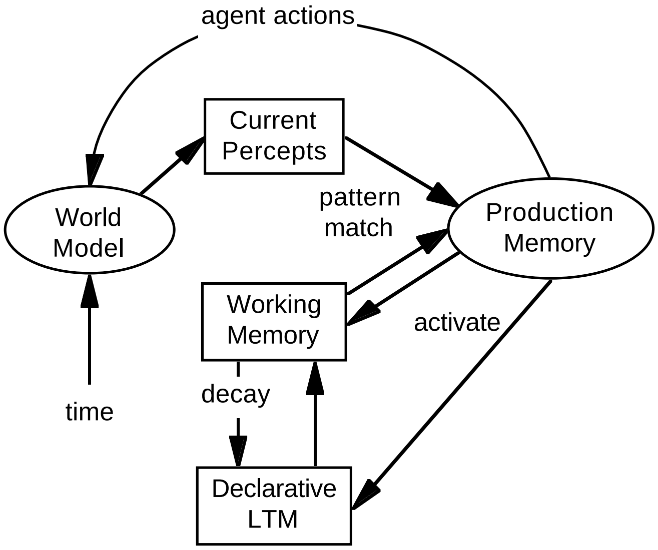

Figure 2. Information and control flow in S-CAPS

S-CAPS, like other production systems, contains a production memory and a working memory. Unlike most other systems, it also contains a declarative long-term memory (LTM) and a world model. The declarative LTM is of little interest in the present discussion; what is critically important is the world model. The world model controls what is available to the agent via perception of the external world.

The first feature to note about S-CAPS is that productions are sensitive to things that are present not just in the internal working memory of the agent. External and internal representations are separate. CAPS is nearly unique among production systems in that working memory can be bounded—it is not of unlimited size. In S-CAPS there are two “pools” of working memory activation: the pool internal to the agent, which is bounded, and the external pool, which is not. It is not the purpose of this paper to go into detail about the capacity limitations of working memory; suffice it to say that there is such a limit, and it adversely affects the performance of the system in terms of both speed and accuracy of performance. Both internal and external representations are available to the pattern-matcher, which does not distinguish between elements in internal and external memory. Thus from the memory perspective, the difference between internal and external representations is the bounded nature of internal working memory.

The second major difference between S-CAPS and traditional cognitive science frameworks is that productions cannot directly affect everything over which pattern matching is done. Productions can alter the activation of elements in the internal working memory, but cannot directly cause changes in the representation of the environment. Instead, productions cause actions to occur. In S-CAPS, actions have a specific form: an action by the agent is an input into the world model. The world model is free to handle that input in whatever way it chooses. Once an action is initiated by the agent, the agent model does not control its outcome—the world model does. The agent's actions can be thought of as requests to the world model for certain outputs, with no guarantee of success.

Another important feature is that the contents of the matching area (the available percepts) can change without the agent doing anything. Note in Figure 2 that "time" is an input into the world model. At the beginning of each production cycle, the world model gets a message or procedure call that some amount of time has passed, and to act accordingly. Thus, the system can be (in fact, must be) responsive to a dynamically changing world.

There are several interesting ramifications of constructing the S-CAPS framework this way. One of them is that the agent-environment system can act without the use of time- and space-intensive cognitive operations. Note that in Figure 2, there is an action path from the world model to perception to productions and back to the world model that does not pass through the agent's LTM or WM. Of course, the agent has to acquire the appropriate productions, but once this has been accomplished such a process can run quickly and fluidly, giving the system a capacity for and providing a clear notion of the idea of "skilled performance." Since little or no "reasoning" takes place during such a process, it should also be fairly rigid.

Another implication of such a framework is that two tasks with identical structure in terms of the problem space (Newell & Simon, 1972) but different representations, one primarily internal and one primarily external, should have different performance characteristics. The reliance on internal working memory should cause the isomorph with the higher reliance on internal representation to be both slower and more error-prone.

Illustrating Integration: S-CAPS at Work

The effectiveness of the S-CAPS framework will be demonstrated with two simulation models of very different tasks, the traditional Tower of Hanoi and a pilot completing a preflight checklist.

Tower of Hanoi

This example is intended to illustrate several of the points mentioned earlier, and will be accomplished through the use of two models of the Tower of Hanoi (TOH). First, the two models will not differ in terms of the knowledge held by the agent, but in the behavior of the world model. Second, they will demonstrate the performance difference between internal and external representation, based on the different world models. Note that this is quite unlike most cognitive science approaches to modeling, where different performance is almost always modeled by differences in the agent's knowledge or capacity.

The Tower of Hanoi is probably the most well-studied puzzle in cognitive science history. The problem space for the TOH is relatively small, having only a few states and simple variants of one operator, moving a disk to another peg. One of the things that has been investigated in the TOH literature is the effect of varying the amount of information represented in different isomorphs of the puzzle. These isomorphs all have the same number of states and operators in the same order, but different surface forms. In general, the amount of time subjects spend on these problems is inversely related to the amount of external representation provided by the isomorph. This is a reasonably powerful and well-known effect (Kotovsky, Hays, & Simon, 1985; Zhang & Norman, 1994).

Strangely enough, this effect has only been modeled once, by Kotovsky and Kushmerick (1991). This model was fairly coarse and, more importantly, it placed the locus of the performance difference as being within the subjects: "[t]he different problems are defined by the subjects' internal problem representation..." This seems counterintuitive, since there are no systematic differences between the subjects in the different conditions—what is different is the puzzle they solved. Thus, the source of the difference ought to be based on different models of the puzzle, not different models of the subjects.

Different environment models form the basis of the S-CAPS simulations. In order to represent subjects working on two isomorphs of the TOH, one set of productions was built to simulate the subjects. The two isomorphs of the TOH modeled here, referred to as "easy" and "hard," differ in their ability to represent move legality. The "hard" version of the world model simply responds to actions and updates the world state. The "easy" version does this and more: it computes and makes available a "constraint:" the legality of moves. This is a reasonable thing for the world model to do—the standard TOH, for instance, may be said to do this. Since an illegal state (a large disk on top of a small disk) is clearly represented in the standard TOH, the world model essentially gives this information to the agent model for "free"—the world model handles the computation of move legality, rather than the agent. This should result in better performance on the "easy" version. (This should, of course, generalize to any two "isomorphic" problem spaces in which one computes constraints for the agent and the other does not.)

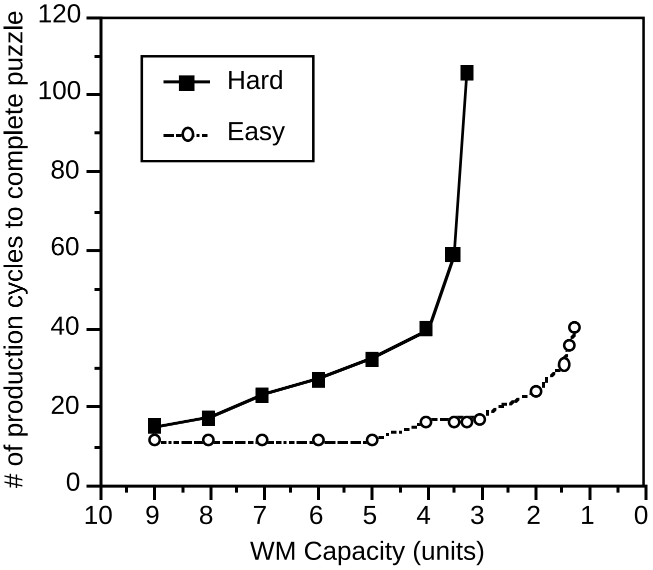

Figure 3. S-CAPS simulation performance

Simulation results were in accord with this prediction. Several simulations were done with both the easy and hard versions of the problems, each with a different amount of activation allowed (i.e. capacity) in the internal working memory. Figure 3 shows that the number of production cycles required to solve the puzzle is clearly a function of both the type of puzzle and the size of the agent's internal working memory. In all cases, the "hard" puzzle did, in fact, take longer to solve than the easy puzzle. In addition, performance on the "hard" puzzle was much more sensitive to the decrease of memory activation available to the agent.

The performance of the simulations at activation capacities greater than nine were the same as those for nine. Simulations were not run at capacities less than those shown on the graph because with smaller capacities the simulations failed to solve the puzzle.

Clearly, the use of external representation has a profound effect on the performance of the simulation, and there is a clear interaction between puzzle difficulty and activation capacity. The increased difficulty of puzzles where the agent must compute move legality is in accord with the empirical results presented by Kotovsky, Hays, & Simon (1985) and Zhang & Norman (1994). Just, Carpenter, & Hemphill (in press) have demonstrated that subjects' working memory capacity is also related to performance on the TOH, and the S-CAPS simulations are consistent with this result.

Preflight Checklist

The second set of simulations will serve to further illustrate many of the same points, but on a more "real-world" task. Human data for comparison is not available for this task, but it serves to highlight the fact that the S-CAPS framework has applications outside of the somewhat overdone TOH puzzle. (These simulations were originally presented in Kirlik & Byrne, 1994.)

The task performed by the pilot modeled is a relatively straightforward preflight checklist. The pilot is required to make a number of safety checks before takeoff, but while working his way through the list, he is interrupted by a critical message from the control tower. He is required to read the message, decide if he can meet the request of the tower, and send a reply. There are three "versions" of the pilot:

Pilot 1. This pilot has little support from the environment, in that he has no copy of the checklist handy and has to maintain the entire list in working memory throughout the task.

Pilot 2. This pilot is different in that he has a copy of the checklist, but nothing that affords checking items off when they are completed—he has no way for the world to tell him where on the checklist he is.

Pilot 3. This pilot has both a checklist and a pen with which to check off items from the list.

Simulation results are presented in Figure 4. Essentially, this figure represents the robustness of the pilots to the interruption. Since Pilot 1 is poorly supported by his environment, his performance degrades most easily. In fact, the first deviation from the initial baseline time represents the pilot having to start the checklist over from the beginning after being interrupted. Further decrements to performance are a results of the system spending production cycles "thrashing" to recover goals and list items.

Pilot 2's performance is better, but the deviation from the baseline time for this pilot also represents restarting the checklist from the beginning. Later degradation of performance with reduced WM capacity is not as poor as for Pilot 1 since list items do not have to be recovered, only goals.

Pilot 3's performance is clearly the most robust in the sense of recovering from the interruption because the environment has afforded him a way to not only represent the checklist, but also his location in it. His performance eventually degrades, primarily due to the difficulty involved in reading the message from the tower.

Thus, these simulations further reinforce the SA position that an important part of understanding complex human behavior is understanding the environment in which it occurs—all of these simulations used exactly the same set of productions to model the pilot.

Conclusions and Implications

Though sometimes characterized as opposing paradigms, traditional symbol processing systems and the situated action perspective can, in fact, be integrated.This is not merely a possibility, but ought to result in an increased understanding of human behavior. By explicitly modeling both the environment and the cognitive agent, it is possible to examine three important sources of variance in behavior:

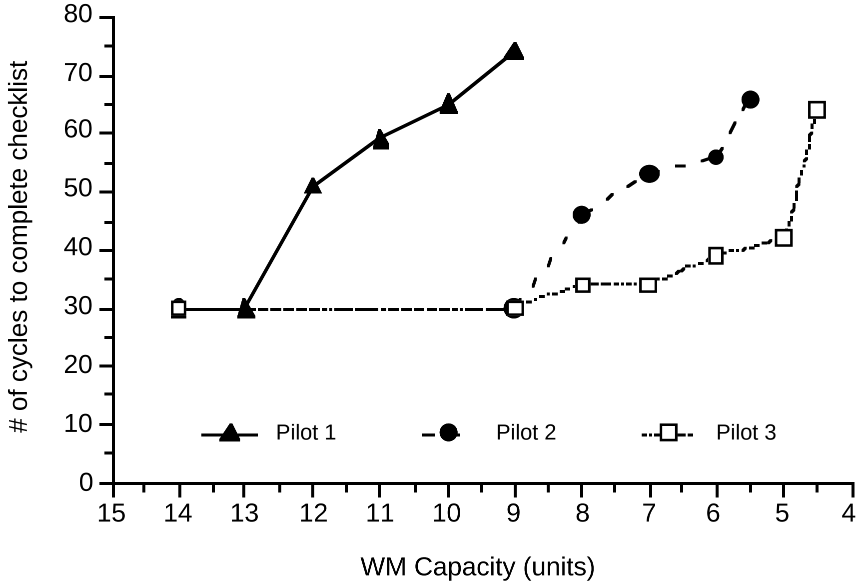

Figure 4. Pilot checklist performance

that which is a result of characteristics of the agent, that which is a results of constraints and affordances in the environment, and that which is a result of the interaction of the agent-environment system. The S-CAPS framework provides a first step toward such integration, and demonstrates, even on the abstract TOH puzzle, the importance of viewing behavior as a situation-cognition interaction.

Clearly, there is much work to be done. We need more formalisms for describing environments, and ways to translate those formalisms into dynamic models with which our simulated agents can interact. While a useful example, the Tower of Hanoi provides a limited picture of the kind of environments that need to be modeled. More interesting results would certainly be provided by supplying a model of a knowledgeable agent in a richer and more dynamic situation, such as an airplane cockpit. Though a task of such complexity may seem a daunting challenge, both the symbol processing community and the situated action community should look upon such an endeavor as an opportunity to enrich each other's perspective, rather than an arena for trying to prove the other camp wrong.

Acknowledgments

I would like to acknowledge the financial support of the National Science Foundation through its graduate fellowship program. I would also like to thank Alex Kirlik for many of the ideas and discussions which led to this paper, as well as comments on an earlier draft. I would also like to thank Sashank Varma for his assistance in modifying CAPS. Finally, I would like to thank the anonymous reviewers for their comments.

References

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

Brunswik, E. (1956). *Perception and the representative design of psychological experiments* (2nd ed.). Berkeley, CA: University of California Press.

Cohen, P. R., Greenberg, M. L., Hart, D. M., & Howe, A. E. (1989). Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine*, *10*, 32–48.

Hammond, K. R. (1986). Generalization in operational contexts: What does it mean? Can it be done? *IEEE Transactions on Systems, Man, and Cybernetics, 16*, 428-433.

Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: An individual differences approach. *Psychological Review, 99*, 123-148.

Just, M. A., Carpenter, P. A., & Hemphill, D. D. (in press). Constraints on processing capacity: Architectural or implementational? To appear in D. Steier, & T. Mitchell (Eds.) *Mind matters: A tribute to Alan Newell.*

Kirlik, A., & Byrne, M. D. (1994). Identifying Environmental Contributions to Skilled Interaction. Presented at the NASA Ames Cognitive Modeling Workshop, February 23, 1994.

Kirlik, A., Miller, R. A., & Jagacinski, R. J. (1993). Supervisory control in a dynamic and uncertain environment: A process model of skilled human-environment interaction. *IEEE Transactions on Systems, Man, and Cybernetics*, *23*, 929-952.

Kotovsky, K., Hayes, J. R., & Simon, H. A. (1985). Why are some problems hard? Evidence from the Tower of Hanoi. *Cognitive Psychology, 17*, 248-294.

Kotovsky, K., & Kushmerick, N. (1991). Processing constraints and problem difficulty: A model. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 790–795).

Larkin, J. H. (1989). Display-based problem solving. In D. Klahr, & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon* (pp. 319–341). Hillsdale, NJ: Lawrence Erlbaum.

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

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

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

Skinner, B. F. (1966). An operant analysis of problem solving. In B. Klienmuntz (Ed.), *Problem solving: Research, method, and theory*, pp. 225-257. New York: John Wiley.

Suchman, L. (1987). *Plans and situated actions: The problem of human-machine communication*. Cambridge, England: Cambridge University Press.

VanLehn, K., & Ball, W. (1991). How Teton blends situated action and planned action. In VanLehn, K. (Ed.), *Architectures for intelligence* (pp. 147-188). Hillsdale, NJ: Lawrence Erlbaum.

Vera, A. H., & Simon, H. A. (1993). Situated action: A symbolic interpretation. *Cognitive Science, 17*, 7-48.

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