COGNITIVE SCIENCE 17, 7–48 (1993)

Situated Action: A Symbolic Interpretation

ALONSO H. VERA AND HERBERT A. SIMON

Carnegie Mellon University

The congeries of theoretical views collectively referred to as "situated action" (SA) claim that humans and their interactions with the world cannot be understood using symbol-system models and methodology, but only by observing them within real-world contexts or building nonsymbolic models of them. SA claims also that rapid, real-time interaction with a dynamically changing environment is not amenable to symbolic interpretation of the sort espoused by the cognitive science of recent decades. Planning and representation, central to symbolic theories, are claimed to be irrelevant in everyday human activity.

We will contest these claims, as well as their proponents' characterizations of the symbol-system viewpoint. We will show that a number of existing symbolic systems perform well in temporally demanding tasks embedded in complex environments, whereas the systems usually regarded as exemplifying SA are thoroughly symbolic (and representational), and, to the extent that they are limited in these respects, have doubtful prospects for extension to complex tasks. As our title suggests, we propose that the goals set forth by the proponents of SA can be attained only within the framework of symbolic systems. The main body of empirical evidence supporting our view resides in the numerous symbol systems constructed in the past 35 years that have successfully simulated broad areas of human cognition.

During the past few years a point of view has emerged in artificial intelligence, often under the label of "situated action" (henceforth, SA), that denies that intelligent systems are correctly characterized as physical symbol systems, and especially denies that symbolic processing lies at the heart of

This research was supported by the Defense Advanced Research Projects Agency, Department of Defense, ARPA Order 3597, monitored by the Air Force Avionics Laboratory under contract F33615-81-K-1539, and by the Office of Naval Research, Cognitive Science Program, under Contract No. N00014-89-J-1975N158.

We are grateful to John Anderson, Susan Chipman, Bonnie John, Jeff Shrager, Jim Greeno, Joyce Moore, Phil Agre, and Lucy Suchman for their comments on earlier drafts of this article. Conversations with Allen Newell contributed much to developing and sharpening our ideas. As always, all of those who helped must be absolved from responsibility for the final product, which is ours alone.

Correspondence and requests for reprints should be sent to Alonso Vera, Department of Psychology, Carnegie Mellon University, Pittsburgh, PA 15213.

intelligence. In fact, SA does not denote a single, sharply delineated position, but a whole congeries of closely related views that share a deep skepticism about the dominant role of symbol systems in the intelligence human beings exhibit, especially in their everyday behavior and in their response to complex or real-time situations.

In this article, we wish to examine whether SA is actually antithetical to symbolic manipulation.To anticipate our conclusions, we find that there is no such antithesis: SA systems are symbolic systems, and some past and present symbolic systems are SA systems. The symbolic systems appropriate to tasks calling for situated action do, however, have special characteristics that are interesting in their own right.

Because there is no official credo to which all those usually associated with SA subscribe, and the points that different authors emphasize are sometimes quite different, we will focus our remarks on the central theme: What is the role of symbol systems in intelligence? Later on, we will sort out and comment upon some of the substrands in the SA literature, but it must be understood that not all of the subprinciples would be accepted by all of those whom we identify with the SA label. If particular SA feet do not fit a particular shoe that we mention, they should not be squeezed into it; no single shoe will fit them all.

Of course, the argument also works in reverse: The symbol-system point of view is no more monolithic than the SA view, and some of us who subscribe to the former view do not always recognize the caricatures of our position that appear in SA critiques of it. However, as suggested before, there does appear to be a central theme that separates the two positions, and we will focus our attention on that.

In the first section of this article we will state briefly what we mean by a physical symbol system, so as to provide a precise template with which we can compare SA systems. In the second section, we will review the accounts of SA that have been given by various of its proponents, taking account of the differences among them and the aspects they single out as salient. In the third section, we will examine some "classical" symbolic systems that were designed for responding, usually in real time, to real or synthetic but complex external environments. We will see how these systems resemble or differ from those designed by the proponents of SA. Then we will compare the symbol systems with some of the more prominent SA systems that have been described and actually implemented. In the fourth section, we will set forth the theoretical conclusions we have drawn from our examination of the two kinds of systems, followed by a brief summary.

1. PHYSICAL SYMBOL SYSTEMS

A physical symbol system is built from a set of elements, called symbols, which may be formed into symbol structures by means of a set of relations.

A symbol system has a memory capable of storing and retaining symbols and symbol structures, and has a set of information processes that form symbol structures as a function of sensory stimuli, which produce symbol structures that cause motor actions and modify symbol structures in memory in a variety of ways.

A physical symbol system interacts with its external environment in two ways: (1) It receives sensory stimuli from the environment that it converts into symbol structures in memory; and (2) it acts upon the environment in ways determined by symbol structures (motor symbols) that it produces. Its behavior can be influenced both by its current environment through its sensory inputs, and by previous environments through the information it has stored in memory from its experiences.

Henceforth, we will usually refer to both symbols and symbol structures simply as "symbols." Symbols are patterns. In a computer, they are typically patterns of electromagnetism, but their physical nature is radically different in different computers (compare the vacuum tubes of the 1940s with integrated circuits of today). And, in any event, their physical nature is irrelevant to their role in behavior. The way in which symbols are represented in the brain is not known; presumably, they are patterns of neuronal arrangements of some kind.

When we say that symbols are patterns, we mean that pairs of them can be compared (by one of the system's processes) and pronounced alike or different, and that the system can behave differently, depending on this same/different decision.

We call patterns symbols when they can designate or denote. An information system can take a symbol token as input and use it to gain access to a referenced object in order to affect it or be affected by it in some way. Symbols may designate other symbols, but they may also designate patterns of sensory stimuli, and they may designate motor actions. Thus, the receipt of certain patterns of sensory stimulation may cause the creation in memory of the symbol (say, CAT) that designates a cat (not the word "cat," but the animal).[1] Of course, this does not guarantee that there is really a cat out there: That depends on the veridicality of the processes that encode the stimulus into the symbol designating a cat. Similarly, a motor symbol may designate the act of "petting" (with some parameters to assure that the cat will be the object of the petting).

The processes that encode sensory stimuli into internal symbols are called perceptual processes, and the processes that decode motor symbols into muscular responses are called motor processes. Perceptual and motor processes connect the symbol system with its environment, providing it with its semantics, the operational definitions of its symbols. Evocation of a symbol

[1] The word "cat" would also be recognized, evoking its own symbol, say *cat*. Evoking" *cat*" can also, by association, access CAT, and vice versa.

by stimuli emanating from the thing or situation it designates also provides the system with access to (some or all of) the information stored in memory about the thing designated. The memory is an indexed encyclopedia; stimuli evoke the appropriate index entries, which point, in turn, to the relevant information.

Symbol systems can be (and sometimes are) used to store in memory representations of external situations. They can manipulate these representations as one way of planning actions, and can then execute these actions to change the external situation. Of course, the internal representation of a real scene will be highly incomplete and may be inaccurate, with the result that the actions may or may not have their desired consequences. We will return to this point later.

Actions also can be, and frequently are, executed without planning. Encoding one or more symbols on the basis of sensory input may trigger the creation of one or more motor symbols, with the consequent execution of the designated action. This sequence would correspond closely to the classical behaviorist stimulus-response sequence, and also to the sequences postulated by SA. As we shall see, the perceptual encoding processes in these instances (and perhaps the motor-decoding processes as well) are likely to be nontrivial.

Finally, sequences of actions can be executed with constant interchange among (a) receipt of information about the current state of the environment (perception), (b) internal processing of information (thinking), and (c) response to the environment (motor activity). These sequences may or may not be guided by long-term plans (or strategies that adapt to the feedback of perceptual information).

Symbolic theories generally make no specific assumptions about what part of the processing takes place at a conscious level and what part is unconscious, except that symbols held in short-term memory (in the focus of attention) are generally available to consciousness, and often can be reported verbally. Hence, the fact that many mental processes are undoubtedly unconscious or subconscious says nothing about whether these processes are symbolic or not. Moreover, "symbolic" is not synonymous with "verbal"; symbolic structures may designate words, mental pictures, or diagrams, as well as other representations of information.

Over the past 35 years, a substantial number of symbol systems have been constructed and tested, successfully, for their ability to simulate human thinking and learning over a wide range of task domains. We cannot review the evidence here; it has already been the subject of a dozen or more books. We have in mind such examples as Newell and Simon (1972), Anderson (1983), Simon (1979, 1989), Langley, Simon, Bradshaw, and Zytkow (1987), and Newell (1990). This is about all that needs to be said about symbols and physical symbol systems for the purposes of this article.

2. WHAT IS SITUATED ACTION?

Proponents of SA refer sometimes to the one, sometimes to the other, of two distinct but related methodologies: a "hard form" and a "soft form." The hard form is a methodology for investigating human-human and human-machine interaction, always within the full context in which they occur. Because all aspects of this context are potentially significant, it is claimed that phenomena must be observed in the actual situation (perhaps using the framework of "ethnomethodology," a term proposed by Harold Garfinkel to refer to "ordinary" people's—as contrasted with specialists'—methods for ordering experience).

Both Suchman (1987) and Winograd and Flores (1986) argue that the methods and terminology of SA should replace current human-computer interaction methods in psychology and AI. To develop better interfaces, they propose, we must focus on how people use them instead of how people think or what computers can do. They do not explain to proponents of the symbolic approaches why the former is antithetical to the latter. We will argue that these concerns are not at all antithetical, but complementary.

The soft form of investigation of SA builds AI systems that incorporate the SA principles of representing objects functionally and interacting with the environment in a direct and unmediated way. The main tangible evidence that permits us to evaluate the claims of SA comes from the attempts that have been made to create intelligent systems that function with little or no planning and minimal representations of their environments. We will be concerned with both hard and soft SA.

Because there is no "official" definition of SA, with various proponents emphasizing different aspects, we will mainly look at what some of its leading proponents, including Winograd and Flores (1986), Suchman (1987), Lave (1988), and Greeno (1989), have written, and analyze some of the programs that are said to illustrate how it actually works. We will evaluate the claim, running through these writings, that SA necessitates an entirely new approach to cognition requiring that humans' interaction with their environment be interpreted nonsymbolically.

We will also comment on variants and flavors of SA that emphasize issues besides the antisymbolic one, particularly questions of context, real-world and social veridicality, and the role of internal representations. Furthermore, the SA approach has, at times, been associated with both Gibsonian and connectionist views. Like the perception of Gibson's (1977) "affordances", the encoding of situational activity is seen by SA as being direct. As in connectionism, SA shares the goal of avoiding a single centralized representation, but instead has independent levels of direct interaction with the situation. The association of SA with affordances and connectionism will be discussed in more detail later.

We will argue that the traditional symbolic view has frequently been misinterpreted in SA work, specifically, in the claim that symbols can denote only linguistic objects and not social and situational conventions. Only a small part of denotation entails direct perception of objects and their behavior, as the denotation of "cat" mainly does. The connections between a symbol structure and its denotation can be complex and highly indirect (e.g., the denotations of concepts like "empirically true," or even "China").[2]

This position must not be interpreted as suggesting that internal representations should be the central focus of investigation in understanding the relation between behavior and cognition. On the contrary, information-processing theories fundamentally and necessarily involve the architecture's relation to the environment. The symbolic approach does not focus narrowly on what is in the head without concern for the relation between the intelligent system and its surround.

A fundamental problem for cognitive modelers is to interleave internal and external states in order to achieve naturalistic behavior. In its extreme form, the SA view argues that there is no need to include internalized world models in the equation. Such internal states, some proponents of this view have said, have no causal effect on behavioral output. The behavior of intelligent systems is fully determined by the contextual situation in which it is elicited. In the following, we discuss the work of several authors who appear to espouse this view, a view that we will contest.

Situated Action and Human-Computer Interaction

"The computer, like any other medium, must be understood in the context of communication and the larger network of equipment and practices in which it is situated" (Winograd & Flores, 1986, p. 5). Winograd and Flores provided one of the first statements of the viewpoint of SA. Although they did not use this more recent terminology to describe their views, they quoted Gadamer on situations:

> To acquire an awareness of a situation is, however, always a task of particular difficulty. The very idea of a situation means that we are not standing outside it and hence are unable to have any objective knowledge of it. We are always within the situation and to throw light on it is a task that is never entirely completed. (Gadamer, 1975, quoted in Winograd & Flores, 1986, p. 29)

Winograd and Flores placed particular emphasis on the difference between acting in ill-structured, real-world situations as compared with well-structured, defined situations, arguing that symbolic approaches, even if they take account of the bounds of human rationality, cannot handle ill-structured situations adequately.

[2] We cannot set forth a whole theory of denotation in this article, but refer the reader to Putnam (1975) and Kripke (1972) for extended treatments of the topic.

> The bounded rationality approach does not assume that a decision maker can evaluate all the alternatives, but it takes for granted a well-defined problem space in which they are located. It is not clear for what observer this space of alternatives exists. In describing the behavior of a manager we (as observers) can formalize the situation by describing it as a set of alternatives with associated properties. In doing so we impose our own pre-understanding to create distinct alternatives out of *the full situation.* In order to write a computer program we are forced to do this kind of analysis a priori. (Winograd & Flores, 1986, pp. 146–147, our emphasis)

This statement is a misrepresentation of the bounded rationality approach, which does *not* assume a fixed, well-defined problem space and given alternatives (see, *contra,* Kaplan & Simon, 1990; Simon, 1973). Winograd and Flores ignored the extensive work that has been done on symbolic systems that continually revise their descriptions of the problem space and the alternatives available to them. We will later describe some systems with such properties.

Starting from the viewpoint of action as situated in complex ill-structured contexts, Winograd and Flores (1986) argued that the most significant challenge facing interface design is to discover the true ontology of human beings with respect to computers: human–computer interaction (HCI). This ontology should be unlike both the one that has evolved from interacting with other (noncomputer) artifacts and the one that has evolved from interaction with humans. Such an ontology is not somehow lying dormant in our minds, but comes into being from our interaction with computers. The design of computers, therefore, requires the study of how humans use them, what they are used for, and what problems are encountered in their use, a claim that we can heartily endorse.

According to Winograd and Flores (1986), designing systems to facilitate work and interaction among humans

> constitutes an intervention in the background of our heritage, growing out of our already-existent ways of being in the world, and deeply affecting the kinds of beings that we are. In creating new artifacts, equipment, buildings, and organizational structures, it attempts to specify in advance how and where breakdowns will show up in our everyday practices and in the tools we use. (p. 163)

As they described it,

> A breakdown is...a situation of non-obviousness, in which the recognition that something is missing leads to unconcealing (generating through our declarations) some aspect of the network of tools that we are engaged in using." (p. 165)

Thus, breakdowns are occasions when the properties of an artifact suddenly become apparent because of a problem either with the artifact itself or with the knowledge of the user.

> The objects and properties that constitute the domain of action for a person are those that emerge in breakdown. (p. 166)

When there is no breakdown, humans are not consciously aware of the properties of the artifacts with which they are interacting.

> In driving a car, the control interaction is normally transparent. You do not think "How far should I turn the steering wheel to go around that curve?" In fact, you are not even aware (unless something intrudes) of using a steering wheel. Phenomenologically, you are driving down the road not operating controls. The long evolution of the design of automobiles has led to this readiness-to-hand. (p. 164)

Like hammers and cars, computers are tools designed by humans for humans.

Winograd and Flores (1986) emphasized that it is a mistake to equate the development of more usable systems with the development of more human-like systems. They believe it is not necessary to create machines possessing genuine intelligence in order for machines to be useful as tools for human activities.

> The key to design lies in understanding the readiness-to-hand of the tools being built, and in anticipating the breakdowns that will occur in their use. A system that provides a limited imitation of human facilities will intrude with apparently irregular and incomprehensible breakdowns. On the other hand, we can create tools that are designed to make the maximal use of human perception and understanding without projecting human capacities onto the computer. (p. 137)

Winograd and Flores see the two central goals of interface design as the anticipation of breakdowns and the creation of tools to resolve them. As they pointed out, breakdowns occur all the time.

> Consider the user of an electronic mail system who tries to send a message and is confronted with an "error message" saying "Mailbox server is reloading. . . ." Mailbox servers, although they may be a critical part of the implementation, are an intrusion from another domain—one that is the province of the system designers and engineers. (p. 165)

Barring breakdowns, *once a domain is well learned* the information is now represented in such a way as to allow processing at a high functional level without the need for conscious awareness of symbolic representations at lower functional levels. Winograd and Flores (1986) stated that "transparency of interaction is of utmost importance in the design of tools, including computer systems" (p. 164). They thus suggested that the goal of the HCI field is to develop machines that are functionally transparent; that is, machines that permit a person to work immediately at a high functional level without needing to know anything about details. Of course, one does not want to learn about hardware to send a letter via electronic mail.

However, only certain kinds of learning can be avoided, for extensive functional knowledge must be acquired by the user who wishes to work at a functional level—a lesson that has been obvious in the computing domain since the earliest development of higher level programming languages.

However, the breakdowns encountered by humans in their everyday lives are not always signaled by domain-specific messages. If I promise I will pick up milk on my way home from work, and my tires start smoking, the smoke signals that I will not be able to get the milk, but it is definitely not a message (or information) in the same domain. The smoking tires provide information from the mechanical domain. The "message" may indicate only that the calipers are frozen, but one consequence is that I will not be able to pick up milk. Intrusions from other domains are common, and dealing with them has certainly played a large role in our ontological history. The ability to resolve breakdowns adaptively is a basic human cognitive capacity.

In addition, some minimum knowledge of the domain-specific constraints imposed by the external system is certainly required. Furthermore, increased knowledge yields the ability to deal with error messages from more distant domains. These are central properties of any adaptive cognitive system, particularly one that is to be capable of dealing with real-world situations, where the richness of context (stressed by the proponents of SA) guarantees that breakdowns will not be rare.

Just as cars are more complex than hammers, and thus require a long learning period to become ready-to-hand (or transparent), so computers are more complex than cars and require proportionately more learning. But for this reason, the enterprise of making computers ready-to-hand or transparent at first contact seems bound to fail, as no other effective human interaction, including human-human interaction, is attained without learning. Pursuing a better ontology might achieve some improvement in first-encounter computer usability, but it is not clear that the way to do this falls out from SA theory.

Notice, also, that the readiness-to-hand of a tool says something about the user's consciousness of the steps in the process of use, but says nothing about whether these processes are symbolic. Symbolic theories have had a great deal to say about the "automation" (i.e., subsidence to the subconscious) of well-practiced processes (e.g., Atkinson & Schiffrin, 1968; Card, Moran, & Newell, 1983); we will return to this point later.

Plans and Representations

Researchers working on HCI from a SA perspective often recommend developing software that allows the user to accomplish complex tasks with little or no planning. As argued before, however, it is unclear whether this sort of direct interaction is possible at all, even in noncomputer environments. A complex task is much more difficult to accomplish when assisted only by direct feedback from the environment than when there is also a way of maintaining some sort of representation of the world. Of course, pure planning, with no situational feedback, is equally ineffective, but it is unfortunate that failures of pure planning schemes have motivated researchers to argue for the opposite extreme instead of a more sophisticated intermediate strategy.

One such researcher who has focused on the issue of planning is Suchman (1987). Planning has traditionally played an important role in systems that interact with the environment. A large part of robotics research, (at least into the 1980s) involved improving robots' plans. Suchman takes the rather extreme position that plans play a role before and after action but only minimally during it.

> I argue that artifacts built on the [cognitivist] planning model confuse *plans* and *situated actions,* and recommend instead a view of plans as formulations of antecedent conditions and consequences of actions that account for action in a plausible way. As ways of talking about action, plans as such neither determine the actual course of situated action nor adequately reconstruct it. (Suchman, 1987, p. 3)

The action is carried out at its own independent level. Before action, plans serve only an organizational or predictive function. Following action, plans serve as "accounts of actions taken" (p. 51). There is no causal relation between the plans and the actions performed by an intelligent system.

> As common sense constructs, plans are constituent of practical action, but they are constituent as an artifact of our *reasoning about* action, not as the generative *mechanism* of the action. (p. 39)

As an example, Suchman suggested that when facing a set of rapids with a canoe, a person would plan a course down the river but this plan would serve no purpose when the rapids were finally run.

> When it really comes down to the details of responding to the currents and handling a canoe, you effectively abandon the plan and fall back on whatever skills are available to you. (p. 52)[3]

This claim is extremely counterintuitive to experienced canoers and to others who perform risky, real-time tasks. One often hears stunt men say that 99% of the work that goes into a stunt is in the planning. This is what keeps them from getting hurt. A person who is likely to be forced to abandon the plan (i.e., is not expert enough to take a canoe through rapids generally following the route mapped out) would be foolish to attempt it. He or she would

[3] Elsewhere on this same page, Suchman (1987) retreated a bit from this strong language, and acknowledged that, even in this kind of situation, the plan may determine initial conditions for the behavior. However, her discussion is, at best, contradictory, and in general, wholly skeptical of planning.

risk plunging over a large waterfall or splitting a head on a rock. Of course, expert canoers or mountaineers sometimes do have to abandon their plans —and sometimes lose their lives.

Nor does Suchman (1987) appear to recognize that most plans are not specifications of fixed sequences of actions, but are strategies that determine each successive action as a function of current information about the situation. In fact, it is precisely the storage of such strategies (and, we would argue, storage in symbolic form) that constitutes the readiness-at-hand of tools. Thus, a good plan for running a rapids includes not only a general path, but also strategies for recovering from deviations.

Planning may be less central to tasks that involve little opportunity for self-harm, but more central to tasks that do not allow easy recovery from error. Training can advantageously substitute preplanning for real-time planning. Pilots' flight training may involve many hours in simulators, with the advantage of putting the student pilots in a greater variety of situations than learning on real airplanes, thus allowing them to acquire and practice strategies for handling real situations. This decreases the amount of planning they have to do in actual flight; but does not decrease their need to have plans and strategies to avoid costly errors.

Along lines similar to those pursued by Winograd and Flores (1986), Suchman (1987) argued for the transparency of "ready-to-hand equipment." With the now familiar example from Wittgenstein, Suchman argued that, although a blind person can be made aware of the physical properties of his or her cane, those properties disappear, in fact the whole object disappears, when the cane is used for its intended function. The blind person is aware only of the shape of the curb that the cane is touching. What the user experiences is the intended function of the artifact. Only if the cane were to break or become bent would the user become aware of its other properties. When objects become nontransparent to users, users become aware of the goal-oriented nature of their activity and equipment.

Phenomenologically, this view seems very plausible. A large body of evidence from psychological investigations of conceptual structure indicates that our concepts of artifacts are strongly shaped by their use (see, e.g., Barsalou, 1987). If someone is asked to list the properties of money, he or she is likely to say it allows you to buy things, it is usually green, it is made out of paper, and so on. It is well documented that some other functional properties such as "flammable" do not frequently appear on such lists. Nevertheless, if you ask people what objects they would try to remove from their house if it caught fire, money is almost always among them. Money seems to become a transparent flammable artifact when the house is on fire, but a transparent material-acquisition artifact when a person is in a department store.

For Suchman (1987), this level of analysis is the appropriate one. Nevertheless, it is also clear from psychological research that there is more to concept structure than what we are consciously aware of. The central issue here is the distinction between conscious and unconscious representation, not phenomenological appeal. This distinction is addressed in detail later.

According to the SA view, when a blind man first begins to use a cane, when a person is first learning how to drive, or when a person first interacts with a particular software application, they have conscious and direct representations of the equipment they are working with. Once these learning tasks are mastered, the equipment "disappears." Proponents of SA would argue that, at this point, the relevant aspects of the situation are no longer in the user's head but in the interaction with the situation. The user is no longer consciously solving a problem or planning: He or she is "simply" doing.

This claim involves a sleight of hand, however. As the task has changed from learning to doing, the information to be processed has changed as well. Information-processing resources are refocused onto the performance of the actual task, which is now less a matter of conscious selective search than it was during the learning period, and more a matter of detection (usually without consciousness) of perceptual cues, and automatic (learned) response to these cues. Both of these processes are symbolic, using the (cue → response) mechanisms usually called productions.

Moreover, the earlier representation of the equipment has not been deleted from the user's memory; it simply need no longer be available to consciousness to do the task at hand. If all the relevant information were not in the user's head, the equipment could never "disappear" in the first place.

Consider further the SA claim that actors are not aware of the tools they are using or the details of their own motions. When entering a curve while driving a car, they are aware of following the road, but hardly of turning the steering wheel, less of moving their arms to turn the wheel, still less of the muscular tensions that produce the arm movements. At the highest level of functionality, the situated action is simply following the road.

Now, in fact, the retinas of the driver are receiving information that is interpreted by an elaborate encoding scheme (but without awareness) as a curve in the road, and the curve is consequently symbolized as such, usually without awareness. This interpretation initiates (neurally, but also without awareness) the symbol emissions that control muscular tensions that cause the arms to move, that cause the steering wheel to turn, that cause the wheels of the car to turn, that cause the car to turn to the left, that cause it to follow the curving road. It is easy to see how the human part of this sequence of events (from receipt of retinal signals to execution of arm movements) can be modeled by a symbolic pattern recognition cum production system. Brooks's (1991) robots, discussed later, are good examples of such systems.

It is incorrect, therefore, to say that situated action of this sort is not carried out symbolically. It is entirely correct to assert that it can be carried out (by an experienced driver) with no conscious awareness of the intermediate links in the chain. Awareness has nothing to do with whether something is represented symbolically, or in some other way, or not at all. It has to do with whether or not particular symbols are available to consciousness in short-term memory. Thus, in an act of recognition, the symbol denoting the object recognized is consciously available, the symbols denoting the features that led to the recognition generally are not. The recognizer is aware of the former but not of the latter (Ericsson & Simon, 1984).

Perhaps the most interesting segment of the chain of events is that which leads, almost immediately, from the interpreted retinal image (curve in the road) to the physical movement (movement of arms grasping the steering wheel). We might represent it like this:

If the road curves to the left—turn to the left.

The language here is purely functional; it surely is not physicalist. It is this functional relation that constitutes the (minimal) internal representation of the situation. A symbol must be stored in memory by the perceptual encoding system to activate the production by satisfying the condition, "if the road curves to the left." Then, the motor response starts with a symbol ("turn to the left") that is transmitted to the motor system, causing the muscles to contract appropriately.

Wittgenstein objected that productions like this would work satisfactorily only if there were a full specification of the states to which they applied. But this claim is clearly incorrect. For example, suppose the road branched, one branch curving to the left, the other continuing straight ahead. What is required to deal with such complications are other, attention-focusing productions that implicitly define "the road." The conditions do not need to be incorporated in the production in question. Similarly, Mother Hubbard, in noticing that the cupboard was bare, did not have to enumerate all the possible foodstuffs that were not in it. To be sure, in the absence of auxiliary, attention-focusing productions, inappropriate action might be taken, as it so often is in everyday life.

The condition in the production we have written is closely related to what Gibson (1977) called an "affordance." Affordances, in the ecological view, are invariants in the environment that are simply "picked up"; this information is perceived directly and requires no processing. For example, in this view it is not necessary to recognize a road in order to perceive that it is "drivable." A road's drivability exists in the perception of the relation between self and environment.

Notice that the affordance is not a simple property of the physical environment: There is no "curving road" out there, and even less a "curving road" on the retina. The functional invariance is produced by an elaborate perceptual process, the kind of process we have simulated in artificial intelligence only with great difficulty (e.g., in Navlab, a driverless mobile road vehicle that we will discuss later). Contrary to Gibson's (1977) view, the thing that corresponds to an affordance is a symbol stored in central memory denoting the encoding in functional terms of a complex visual display, the latter produced, in turn, by the actual physical scene that is being viewed.

In the same way, there is in the environment no action of "turning to the left." There is only a symbol initiating a sequence of muscular processes that propagate the action through the mechanism of the car into the scene, an inverse decoding as complex as the encoding on the perceptual side. The action of the production is the symbol that initiates this whole sequence: denotes it and its functional outcome of following the road. The complexity of this perceptual encoding and motor (and environmental) decoding is vividly felt in driving when one rounds a curve that has a variable radius of a curvature.

Thus situated action cannot get along without an internal representation. In fact, its representation is the result of a complex translation into functional language of a physical situation of which the functional significance is only implicit. What makes the translation especially useful is that the resulting representation is extremely simple, linking simply encoded, complex situational clues to simply encoded, complex motor responses. In early AI problem-solving systems, GPS, for example, this linkage was stored in a "table of connections," with only the difference that the "motor action" modified an internal problem representation rather than the external one. As we shall see presently, this difference is not consequential. What is consequential is the prelearning that reduces an extremely complex sequence to a learned production—an affordance.

The significance of the "functionalism" of the environment in relation to actions is revealed by an interesting feature of natural languages, which often "verb" nouns and "noun" verbs. Consider the verb "wash" in English. In means to cleanse something with water. The noun "wash" means a set of articles set aside for washing or in the process of being washed. We can represent the relation by the production:

If O is wash → wash O.

Now the "wash" in the condition of this production is a symbol produced by an elaborate perceptual coding process that recognizes a pile of soiled and rumpled objects as calling for cleansing. The "wash" on the action side of the production is a symbol denoting the action required. That action, of course, will ordinarily be an elaborate program, in our culture, requiring the collaboration of automatic washers and dryers.

To say that the representation, when action is problematic, is functional, does not make it less a representation, nor deny its symbolic character. It is precisely this symbolic representation that provides the "readiness-at-hand" of the tool. And when there is "breakdown"—when situations become problematic, that is, less than transparent—the representation is quickly elaborated down to the levels of detail required for diagnosis, problem solving, and repair. In the completely unproblematic situation, the actor will be aware only of the highest, simplest functional representation. Awareness will expand to lower, more procedural, and "physicalist" representations as soon as problems arise. If lack of experience and knowledge or perceivable information in the situation prevents this expansion, the problems will not be solved and action will fail. (Not all drivers round all curves successfully.)

A functional description of the world (i.e., a description in terms of something like affordances) is one that allows simple mappings between our functional models of what is out there (e.g., road curves to the left) and our functional actions (e.g., turn to left). However, the resulting simplicity of the relation between these two functional representations does not imply that the relation is somehow "direct" or unmediated. It is, in fact, complexity of mediation (in the form of many representational layers) that affords this simplicity. Simplicity, in turn, gives the relation the phenomenological character of being direct. Affordances are in the head, not in the external environment, and are the result of complex perceptual transduction processes.

Suchman (1987) posed the goal of the SA approach as follows:

> What motivates my inquiry...is not only the recent question of how there could be mutual intelligibility between people and machines, but the prior question of how we account for the shared understanding, or mutual intelligibility, that we experience as people in our interactions with others whose essential sameness is not in question. (p. 6)

Presumably, this human–computer mutual intelligibility entails having a medium of interaction that is largely transparent because this is how we conceive human–human interaction to be. Although HCI is not expected to develop along exactly the same ontological lines, we do want it to meet the same criteria. We have argued that although these are excellent goals for computer developers, the problems of achieving them cannot be divorced from issues of symbolic representation, learning, planning, and problem solving.

Pedagogical Issues

There are at least two rather disparate directions that the hard SA approach has taken. Although those following both directions consider themselves to be closely related, only one is in essential conflict with the symbolic approach. One group, as discussed previously, has followed Winograd and Flores (1986) in arguing that information-processing theory and methodology cannot account for behavior in general, and especially in tasks that involve direct and continuously changing interaction with the environment.

The other group, exemplified by such researchers as Jean Lave and Jim Greeno, argue that there is a problem in the relation between the way things are taught in school and their application to the real world, and that because of the unfortunate divorce between education and real life, our educational system is ineffective in preparing people for real-world problems. The only way for knowledge to apply to the real world, they claim, is for learning to involve doing real-world problems in the first place.

Lave (1988) proposed that what has been traditionally understood as the *transfer of learning problem* should be recast in situated terms. Based on her experimental findings, Lave

> recommend[s] a move away from..."learning transfer" as the explanation for cognitive continuity across contexts, to an analytic approach in terms of the dialectical structuring of the activity of persons-acting in setting. (p. 19)

Dialectical structuring entails the interaction of components, the existence of each being contingent upon that of the others.

> It is not at the level of activity, but at the level of a set of transformations of articulated structuring resources that activity may be said to be "the same" from one occasion to the next. This helps to explain why transformational relations which are part of "intentionless but knowledgeable inventions," can be anticipated and expectable without having literally been experienced as the resolution shapes in relation with which experience is constituted. (Lave, 1988, p. 189)

Thus, in her view, learned "facts" do not transfer from one situation to another. Continuity is provided by a set of dialectical relations:

> Like "rationality," the continuity of activity over contexts and occasions is located partly in the person-acting, partly in contexts, but most strongly in their relations." (p. 20)

Greeno (1989) argued along the same lines as Lave (1988), that in real-life situations, symbolically represented knowledge does not translate well into useful skills. There is a gulf between abstract symbolic knowledge and real objects in the world, and mental models aren't available as pedagogical tools, because they are not public. Taken as a proposal for changing education, this argument suggests that special efforts must be made to bring symbolic knowledge domains (e.g., physics and math) into closer alignment with real-world objects, for example, by using physical models to provide semantic interpretations of symbolic formulas and algorithms.

A frequently cited anecdote used to demonstrate the point is the cottage cheese problem (in de la Rocha's chapter of the 1984 book edited by Rogoff & Lave). A person is required to take 3/4 of 1/2 of a cup of cottage cheese. Instead of multiplying the two fractions and then trying to measure 3/8 of a cup, the problem solver first fills a cup, removes half, spreads it out into a circle, and then cuts out a quarter. Greeno agrued (1989) that physical models having component objects that correspond closely with those found in real situations are better pedagogical tools than symbolic formulas and algorithms. Does this argument imply that symbolic knowledge does not underlie the central processes of ordinary everyday cognition? We think not.

There is no reason to believe that knowledge about interaction with real-world objects is not symbolically represented. A particular symbolic representation does not by itself guarantee connection to all other symbolic knowledge. Therefore, the fact that formal symbolic knowledge is often not transferable to analogous real-world problems is not a challenge to symbolic theories. Explicit symbolic representation is not incompatible with the vagueness, immediateness, or variability across time that are required to represent the external world. Asserting such incompatibility is perhaps the most significant of the misrepresentations in SA's description of the symbolic approach.

Lave's (1988) and Greeno's (1989) pedagogical concern is that skills acquired in formal school settings often do not operate in the real-life situations for which they pretend to prepare the student. Although Lave wished to reformulate the question, this is nevertheless the classical and well-known problem of transfer of learning. It is a fundamentally important problem, which calls for continuing and expanded study, but has nothing to do with the adequacy of symbolic systems as theories of intelligent action, in schools or in the real world. It is not important whether the problem is or isn't correctly labeled as "transfer." Whatever it is called, the problem is as frequently addressed in the learning literature within the symbolic tradition as in the SA literature.

If the SA view is suggesting simply that there is more to understanding behavior than describing internally generated, symbolic, goal-directed planning, then the symbolic approach has never disagreed.

> The proper study of mankind has been said to be man. But. . .man—or at least the intellective component of man—may be relatively simple;. . .most of the complexity of his behavior may be drawn from his environment, from his search for good designs. (Simon 1969, p. 83)

3. REPRESENTATION OF SITUATED ACTION IN SYMBOLIC PROGRAMS

The significance for cognitive science of the SA view can be evaluated by observing current research in AI. If SA has led researchers down novel paths, which cannot be followed along traditional information-processing lines, then it can be judged successful, even revolutionary. If, on the other hand, it has produced few innovations and those it has produced are compatible with the traditional information-processing framework, then its scientific contribution may be modest.

In this section we will review some research projects that, collectively, represent both sides of the story. First, the Phoenix system and Navlab will be discussed. Both of these projects are grounded in traditional symbolic theory, but attempt to deal with behavior in real-time, highly interactive, and changing environments. We then take The Tower of Hanoi problem, which has been a basic AI task, and explore how it might be performed along SA lines. This exploration is followed by an account of Larkin's (1989) "display-based" system, DiBS, which exhibits other features of SA within a thoroughly symbolic structure. These four examples will provide, collectively, a test of whether symbol systems can act intelligently in circumstances where limited computation resources must cope with real-world complexity in real time. Of course, there are many other examples we could have used.

With the experience gained from these efforts to relate symbolic with SA theory, we then look at two current research projects that have been viewed as exemplifying SA principles: Brooks's (1991) creatures and Agre and Chapman's (1987) Pengi. These two programs appear to be the principal examples, to date, of actual implementations of the SA philosophy. Because running programs provide the constructive demonstrations of feasibility and sufficiency that are missing from verbal philosophizing, these programs deserve our close attention. We will be especially interested, first, in comparing them with the symbol systems described earlier, and second, in judging whether they are, in fact, nonsymbolic in character.

The Phoenix Project

Cohen, Greenberg, Hart, and Howe (1989) developed Phoenix, a highly interactive intelligent system that functions in a complex realistic environment. The Phoenix environment simulates forest fires in Yellowstone National Park. The simulated fire-fighting system disposes of bulldozers, crews, and other equipment that it can deploy to control the fire. One central goal of the project is "to understand how complex environments constrain on the design of intelligent agents" (Cohen et al., 1989, p. 34). The aim is to define a set of general design principles for intelligent agents based on the behavioral, environmental, architectural, and task parameters that define the *context* of the task.

In the Phoenix environment, the fire simulator creates an accurate representation of most types of forest fires. Cohen et al. foresaw that their use of a simulated environment might be challenged by proponents of SA. "In response to the criticism that simulators can never provide faithful models of the real, physical world, we argue that the fire environment is a real-time, spatially distributed, ongoing, multi-actor, dynamic, unpredictable world" (p. 36). Arguing that the fire environment does not represent a legitimate testing ground for SA would be analogous to suggesting that a pilot in a flight simulator is not acting situatedly because the experience is being artifically c-eated.

The fire-fighting side of the system has a multilevel design where different agents including the fire itself act independently of one another. The bulldozers, for example, have basic reflexes that keep them from being damaged. This allows them to function autonomously when they are threatened by the fire, and does not require real-time intervention from the planner in order to help them.

There are two significantly different time scales on which events occur in the Phoenix environment. There is planning and plan execution, which can take on the order of a few hours, and there is a reflexive level, which requires reactions measured in seconds (e.g., for a bulldozer to avoid being damaged by encroaching fire). These two levels are referred to as the cognitive and reflexive components, respectively. The two components are virtually independent of one another in their instantiation.

Each Phoenix agent has its own cognitive and reflexive components. Along with agents such as bulldozers, crews, and airplanes, there is also a Fireboss, which, unlike the others, has neither sensors nor a reflexive component. The Fireboss is the only agent with a static map of the whole park. However, all other information about the environment available to the Fireboss is received through other agents.

The cognitive component of the system consists of a plan library, a time line, a cognitive scheduler, and a state memory. The cognitive scheduler applies the actions required to execute a selected plan. The time line is simply an explicit representation of the temporal relations among plans, actions, and external events. The state memory holds information about equipment availability, weather conditions, and sensory data (including feedback about unplanned reflexive actions). The cognitive component does not become aware of state discrepancies caused by reflexes or unexpected errors until routine status-checking actions are executed.

For example, if a plan indicates that two bulldozers should rendezvous at one side of the fire, then the cognitive scheduler attempts to execute the actions necessary to achieve this, based on the information in state memory (e.g., information about the current location of each bulldozer). If one of the bulldozers were suddenly forced to flee the fire because of changing wind conditions, then the action planned by the cognitive scheduler would not succeed. The actions necessary to execute the plan need to be reevaluated in the new context once the bulldozer's new location becomes available in the state memory.

This ability to reformulate actions required to execute a plan, or even to select a different plan if necessary, is one form of context sensitivity and error recovery demonstrated by the Phoenix agents. Although the Phoenix system does not yet incorporate learning, future work is aimed at having the system learn patterns of frequently performed reflexes as well as short plan segments.

Cohen et al. (1989) indicated that the motivation for creating a system consisting of two almost independent functional components was the belief that a purely reactive system based only on reflexes would be unable to perform Phoenix's complex task.

> Although some researchers have suggested that longer-term plans can emerge from compositions of reflexes (Agre & Chapman, 1987; Brooks, 1986), we do not believe that compositions of reflexes can handle temporally extensive planning tasks such as resource management or spatially extensive tasks such as path planning with rendezvous points for several agents. Thus, we have adopted a design in which reflexes handle immediate tasks, and a cognitive component handles everything else. (Cohen et al., 1989, p. 40)

A discussion of the work of Brooks (1991) and Agre and Chapman (1987) follows after the description of the Navlab symbolic system and a situated interpretation of the Tower of Hanoi problem.

The Navlab System

Navlab (Thorpe, 1990) is a robot vehicle with independent perception, actuation, and decision systems that act in real time to move it to different locations while avoiding harm to self or others. One part of the project involves navigating autonomously through a suburban neighborhood. This has been successfully achieved for a half-mile course that includes three intersections. This route was traversed at a speed of 8 to 10 miles per hour, although the Navlab has subsequently shown its ability to proceed at more than 40 miles per hour on a public highway.

The vehicle uses a variety of sensors including a scanning, laser rangefinder, sonar, a video camera, and an inertial navigation sensor. Although these systems are not always used at the same time, they yield distinct but somewhat overlapping information. For example, both the laser and video camera can be used for landmark detection. Landmarks are used to determine the vehicle's exact location in the world, which can also be determined by the inertial navigation sensor.

Navlab needs to be able to determine where it is, where it wants to go, and how to get there. The first question represents the most important function of the system because the other two depend on this information. Data from the sensors is integrated by a *state maintenance system* and this status information is placed into a buffer shared by other systems. The information in this buffer changes in real time as the vehicle's state and position change.

Independent modules address the questions of where to go and how to get there. These modules act as "clients" to the controller. The Navigator module determines the vehicle's motion based on the status inforamtion collected by the Maintenance system. Observer modules, on the other hand, do not affect the motion, but do things like building maps based on the status information provided by the Maintenance system.

The actual physical behaviors of the system are generated by the Actuation system. This system acts directly on external software that effects behavior. Navlab also has a system that interrupts when an action proposed by a system or client is considered to be potentially hazardous. For example, there is an obstacle detector that overrides the activity of all other modules when an obstacle is found. A high-level *Arbitrator* manages the interaction among the sensors, actuation systems, and other modules.

As mentioned before, information from all the vehicle's physical sensors are integrated into one piece of information by the state processor. This one piece of data represents the location of the vehicle in terms of both its orientation in three dimensional space and its dead-reckoned position. There is also a second "perceptual" system that uses the information gathered by the sensors about the vehicle's location but otherwise functions independently. These are the modules that decide where to go and how to get there. There are thus two perceptual systems, one for determining location and one for creating maps that the Navigator follows. These systems chain off one another (because the data are continually changing when the vehicle is in motion), but remain functionally and physically independent.

In one of its configurations, the Navlab uses a neural net to navigate its way around. The net is trained by matching the input from a video camera focused on the road ahead with the motor behavior of a human driving the vehicle. This type of network has been very successful at navigating the vehicle on real roads. In order to get from one location to another, however, the vehicle must plan a path to the goal location and then follow that plan. This plan is based on an internal map of the area (e.g., a suburban neighborhood). Symbolic knowledge about the dead-reckoned location of the vehicle with respect to the map must be integrated and used in order to reach the goal. When the vehicle needs to make decisions about how to get to its goal, the Arbitrator overrides the network and allows the Navigator to drive.

> The neural network driving modules are good at reactive tasks such as road following and obstacle avoidance, but the networks have limited capability for the symbolic tasks necessary for an autonomous mission. The system of networks cannot decide to turn left to reach a goal. After making a turn from a

> one-lane road to a two-lane road, the system does not know that it should stop listening to one network and start listening to another. Just as a human needs symbolic reasoning to guide reactive processes, the networks need a source of symbolic knowledge to plan and execute a mission. (Pomerleau, Gowdy, & Thorpe, 1991, p. 281)

Navlab contains systems that detect unforeseen problems (internal or external) and can execute the appropriate behaviors to avoid or correct them. Because all these systems interact in real time, both the state information and the plans based on them are in a constant state of change. Nevertheless, the vehicle can achieve its goals successfully. It is able to move around to planned locations without damage to itself or others.

We see that Navlab is a symbolic system[4] that successfully combines capabilities for quick response to the environment with strong planning capabilities to handle events that are nonlocal in time or place. Like Phoenix, both its planning actions and its behaviors are highly dependent on context.

Equivalence of Strategies

One presumed major distinction between schemes for SA and conventional symbolic AI systems, often emphasized by proponents of SA, is that the conventional systems perform tasks by means of internal planning with a model of the real-world situation, whereas SA schemes interact "directly" with the situation. Because this contrast is frequently mentioned in the SA literature, but nowhere formalized, any attempt to test its validity must first propose some criteria of evaluation.

One indication that a system is doing little or no planning is that it does not hold in memory, during performance of the task, either an explicit action plan (at some level of generality or detail) or an elaborate structure of goals—a goal stack, say. SA is not supposed to require a representation of the real-world situation being acted upon. This would imply that the conditions of its productions will be features of the external situation rather than an internal model of it.

We consider a well-known "toy" task, the Tower of Hanoi problem, which is presumably ideally suited for planful solution, and we show that a symbolic system can solve it in a manner that satisfies the conditions for SA (Simon, 1979). First, we will describe a common strategy for solving the Tower of Hanoi problem, which Winograd and Flores (1986) would undoubtedly characterize as "rationalistic." Then we describe an alternative "perceptual" strategy, which uses neither an internal representation nor a goal stack. Third, we will report evidence on how human subjects solve the

[4] The question may be raised of whether the network component of one version of Navlab is symbolic. We believe that it is, but will not undertake a discussion of this issue here.

problem. Finally, we will describe a more general system, DiBS (Larkin, 1989), which uses a perceptual strategy to perform a variety of tasks.

The Tower of Hanoi puzzle involves three vertical pegs and a number of doughnut-like disks of graduated size that fit on the pegs. At the outset, all the disks are arranged pyramidally on one of the pegs, say A, with the largest disk on the bottom. The task is to move all of the disks to another peg, C, say, under the constraints that (1) only one disk may be moved at a time, and (2) a disk may never be placed on top of another that is smaller than itself.

The Goal Recursive Strategy. The pyramid of disks can be moved from A to C in the following three stages: (1) the pyramid consisting of all save the largest disk is moved from A to the other peg, B; (2) the largest disk is moved from A to C; (3) finally, the pyramid on B is moved to C. Only the second stage, of course, corresponds to a legal move. The first stage, which clears A and C for Move 2, and the third stage, which brings the remaining disks to C, are themselves Tower of Hanoi problems with one less disk than the original problem; hence they can be solved by decomposing these subproblems recursively into the same three stages.

A symbolic production system consisting of six productions can solve this problem without view of the physical situation, and holding in memory only (1) a stock of the initial goals and the additional goals it has generated en route and not yet achieved, and (2) a working-memory symbol that describes the status of the current goal (feasible, infeasible, solved). This production system, whose conditions refer entirely to tests on the internal working memory, including the goal stack, is shown in Figure 1.

If this recursive system is allowed to announce its moves instead of making them physically, it does not require a full representation of the current situation of pegs and disks, but only knowledge of the current state of the goal stack and the status of the current goal. Because it does not require a goal stack, we would not describe its action as "situated." It is especially not situated because it uses no direct information at all about the changing external situation. Of course, by the same token, if, unbeknownst to it, any change were made in the external situation by the experimenter or a third person, the system would be in deep and incurable trouble. It has no sensitivity to contexts that could change the situation.

The Perceptual Strategy. By introducing some "perceptual" productions, whose conditions are tested against visible features of the external, real-world Tower of Hanoi, an alternative production system can be built that dispenses with the goal stack. The behavior of the system is steered entirely by perceptual interaction with the changing configuration of disks in the puzzle.

Goal Recursion Strategy

```
P1. State = Problem-solved -> Halt
P2. State = Done,
    Goal = Move the pyramid consisting of the k smallest disks to Peg A -> Delete (STM),
    Goal <- Move the pyramid consisting of the next k smallest disks to Peg A
P3. State = Can, Goal = Move the pyramid consisting of the k smallest disks to Peg A
    -> Delete (STM), Move disk k from its peg to Peg A
P4. State = Can't,
    Goal = Move the pyramid consisting of the k smallest disks to Peg A -> Delete (STM),
    Goal <- Move the pyramid consisting of next k smallest disks to the peg other than Peg A
P5. Goal = Move the pyramid consisting of the k smallest disks to Peg A
    -> Test (Move disk k from its peg to Peg A)
P6. else -> Goal <- Move (Pyramid (n), Goal-peg)
```

Perceptual Tests

```
Test (Move disk X from its peg to Peg A)
T1. If for all disks Y, Y is on the goal-peg, declare the problem solved
T2. If disk X is on Peg A, declare the current goal to be done
T3. If disk X is the top disk on its peg, and is free to move to Peg A,
    declare that the desired move can be done
T4. else -> declare that the desired move cannot be made
```

Figure 1. Production systems for goal recursion strategy and perceptual tests.

First, it notices the largest disk that has not yet been placed on the goal peg. If it sees that this disk can legally be moved to the goal peg (there is no smaller disk above it or on the goal peg), the system makes that move. Else, the system notices the largest disk (on source or target peg) that is impeding the move, and sets the goal of moving this blocking disk to the other peg. This procedure is repeated until a move can be made. Then a new "largest disk" is noticed again, and the cycle repeated until the problem is solved. The production system is displayed in Figure 2.

When it uses the perceptual strategy, the system does not need to retain a goal stack. It can always determine what move to make next by observing the actual current state of the puzzle and replacing its previous goal by a new one. Moreover, if there is any external interference with the puzzle, or if the system leaves it partially unsolved and returns to it later, it can resume its activity without any reference to memory or any difficulty, and continue to solution. Finally, the strategy still works if the task is generalized to allow different starting and goal situations.

The perceptual strategy for the Tower of Hanoi would seem to meet all of the criteria for SA. The system need not construct in memory a representation of the situation or a goal stack. It reacts appropriately to any change in the external situation, whether due to its own actions or to the interven-

Perceptual Strategy

```
P1. State = Problem-solved -> Halt
P2. State = Done, Goal = Move disk k from its peg to Peg A -> Delete (State), Delete (Goal)
P3. State = Can, Goal = Move disk k from its peg to Peg A -> Delete (State),
     Move disk k from its peg to Peg A
P4. State = Can't achieve goal because of disk J ,
     Goal = Move disk k from its peg to Peg A -> Delete (STM),
     Goal <- Move disk J from its peg to the peg other than Peg A
P5. Goal = Move disk k from its peg to Peg A -> Test (Move disk k from its peg to Peg A)
P6. State = Biggest (J) -> Goal <- Move disk J from its peg to the Goal-peg
P7. else -> Test (Biggest-remaining)
```

Perceptual Tests

```
Test (Move disk X from its peg to Peg A)
T1. If for all disks Y, Y is on the goal-peg, declare the problem solved
T2. If disk X is on Peg A, declare the current goal to be done
T3. If disk X is the top disk on its peg, and is free to move to Peg A,
     declare that the desired move can be done
T4. else -> identify the largest disk blocking the movement
Test (Biggest-remaining)
T5. identify largest disk not on goal-peg
```

Figure 2. Production systems for perceptual strategy and perceptual tests.

tion of another. It can take up the task from any situation and is undisturbed by interruptions. At the same time, it is clearly a symbolic information-processing system, demonstrating that such a system can carry out situated action.

For later comparison with a SA system, Pengi, we should take note of the perceptual predicates that this Tower of Hanoi strategy uses. The most important of these are (A) "largest disk not yet on goal peg," and (B) "largest blocking disk." The latter is the larger of (a) the largest disk above the one to be moved next, and (b) the largest disk on the goal peg of the disk to be moved next.

Note that these predicates are defined in functional, not physicalist, terms; they denote affordances. The predicates name disks in terms of where these disks are situated in relation to other disks of interest, and hence are perceptual predicates *par excellence.* The system is defined so that disks having these properties will be noticed "immediately," that is, in real time. Finding them is accomplished very efficiently by surveillance of three pegs in specified locations. Hence, the perceptual scheme defined for this Tower of Hanoi strategy is very similar to the SA scheme defined by Chapman (1989) to demonstrate how a system like Pengi could solve the fruitcake problem originally posed by Nilsson (1988; see the discussion of Pengi in a later section of this article).

Human Behavior. A great many human subjects have been observed working the Tower of Hanoi Problem and various isomorphs of it. The successful subjects have used a substantial number of different strategies. In particular, variants of the perceptual strategy are frequently used by subjects new to the problem, but as they become more skilled in solving it, they tend to move toward the recursive stragegy (e.g., making plans to move whole "pyramids," and storing them). Here we see a gradual shift *from* situated action to more planful action (see Anzai & Simon, 1979).

Many subjects also number the disks, from small to large, and develop a rhythmic pattern for remembering the sequence in which they should be moved: 1 2 1 3, 1 2 1 4, 1 2 1 3, 1 2 1 5, etcetera. The sequence is not memorized but is produced by a generative rule that is memorized: Disk 1 is moved on every other step, Disk 2 on half the remaining steps, and so on. This strategy is, of course, vulnerable to interruption and outside interference, and can hardly be regarded as situated action. It is the strategy most frequently learned independently by subjects.

A General Perceptual System

Jill Larkin (1989) constructed the DiBS system to illustrate the general properties of what she called "display-based" (i.e., situated) problem solving. She demonstrated how DiBS solves a linear algebraic equation, how it solves the Tower of Hanoi problem, and how it would brew coffee.

> The model is a production system. . .together with a working memory. Conceptually, the computer-implemented working memory is divided into two kinds of elements: those reflecting external real-world objects and those reflecting elements held internally in the solvers' short-term memory. . . . [T]he model acts when the conditions of some production are satisfied by the contents of working memory, here by a combination of external objects and internal items. The associated actions then change working memory. Again these changes can either be changes to the physical world (e.g., pouring water from a carafe), or to the internal world of the solver (e.g., setting the subgoal to get coffee beans). When working memory has been altered in either of these ways, ordinarily the conditions of some new production are satisfied, and its actions are then implemented. Cycles like this repeat until the problem is solved or until no production is satisfied. . . . DiBS reflects the following mechanism: When a solver looks at a display, various visible objects suggest or cue information about where they ought to be placed in order to solve the problem. In assembling a coffee maker, one knows the equipment and has an internal memory tag cued by each object indicating where it should go. . . .
>
> Sometimes DiBS cannot move an object to the place where it "wants" to be. . . . [D]ifferent problem situations require different knowledge about how to get rid of offending blockages, but the same basic mechanism applies to all three, and certainly could readily be extended to other tasks involving assembling physical objects, or manipulating symbol arrays. . . .

> A central feature of all the DiBS solutions discussed is that very little information must be held in internal working memory. Many attributes of the data structure are attributes of external objects. These need not be stored internally but can be observed from the environment. Others are associated directly with an object, so that the object may well serve as a helpful cue to remembering the attribute. (p. 323)

Larkin observed human errors in performing the tasks that DiBS does. Of 100 errors observed in coffee making, 70 occurred because the display concealed crucial information about the state of the system (e.g, whether the coffee pot was empty or already full). A second cause of error was lack of knowledge of how to assemble or dissemble some piece of equipment, or other knowledge of operations essential to performing the task. These are precisely the kinds of errors we would expect in display-based, or situated, problem solving.

Brooks's Creatures

An interesting application of SA views was proposed by Rodney Brooks. Although Brooks preferred not to be associated with a particular theoretical label, his aim was to develop a system with decentralized representation that can function in the real world. We can ask whether the principles that underlie his representation are genuinely different from those of symbolic theory, as he claimed.

Brooks (1991) argued that AI should focus its energy on the development of functionally complete intelligent systems (sometimes referred to as broad agents). Instead of developing models of particular, isolated, human-level capacities, AI research should work toward systems that can operate independently in the real world, both sensing and acting on it. These systems need not demonstrate human-level intelligence in any particular domain, but they should be able to coexist autonomously in a human environment. Intelligence should be built up incrementally by adding new control layers to simple systems that already work.

To this purpose, Brooks (1991) built mobile robots that are able to wander around and explore a normal office environment. These "Creatures," as he called them, have a number of functionally distinct control layers that act independently on the environment. The "higher" layers have supervenience over the lower ones and can take over their functions as a way of achieving more complex behavior while maintaining lower level reactions to the environment (e.g., going to investigate distant objects while still avoiding obstacles on the way there).

The Creatures' sensors feed directly into distinct activity layers, each of which can react to the input with its own set of motor behaviors. The system does not, at any point, have a centralized representation of its world. Each layer has only the information about the environment that it requires and this information is processed independently and in parallel. Information is not kept after it is used in service of an action. The Creatures' representations are temporary and distributed.

The information passed by the sensors to the layers and from one layer to another is basically numerical. The lowest layer receives data from a number of sonars. Its purpose is to keep the Creature from hitting objects. "[The lowest layer] simply runs the sonar devices and every second emits an instantaneous map with readings coverted to polar coordinates. This map is passed on to the collide and feelforce finite state machine." (Brooks, 1991, p. 153). The Collide machine looks for objects which lie in the Creature's direct path. A halt command is sent to the machine in charge of forward motor behavior if the polar coordinates indicate the presence of an object dead ahead. The purpose of the second layer is for the Creature to wander around. The third layer causes exploratory behavior.

Information from the sensors is not always used in its "raw" form, however. "The contributions of each sonar are added to produce an overall force acting on the robot. The output is passed to the *runaway* machine which thresholds it and passes it in on to the *turn* machine which orients the robot directly away from the summed repulsive force." (Brooks, 1991, p. 153). Brooks's Creatures are very good examples of orthodox symbol systems: Sensory information is converted to symbols which are then processed and evaluated in order to determine the appropriate motor symbols that lead to behavior.

The goals of each layer are integrated by sending messages from higher to lower layers, either overriding or competing with messages generated in the lower layer. Brooks (1991) referred to these two mechanisms as suppression and inhibition, respectively. Messages from higher layers affect specific machines in lower layers. These relations are carefully set by establishing specific topographical connections between machines. Messages from higher layers can be ignored if the particular lower layer machine that would enact the behavior is busy doing such things as avoiding obstacles.

This system worked well enough that Brooks set as his goal a robot with insect-level intelligence by 1990 (the work for the article cited was done in 1987). There has been no report that the goal was reached. Although the current level of performance represents an impressive application of SA (but not symbol-free SA), it is still problematic whether this approach will extend well to more complex problem solving. Noncentralized representation and planless action may work adequately for insect-like creatures, but it may not suffice for higher level problem solving.

Surely, Simon's (1969) ant does not need (and almost certainly does not have) a centralized and permanent representation of its environment. To navigate its zigzag way home, the ant does not make use of a representation of the location of each grain of sand in relation to its goal. It deals with each obstacle as it comes to it and does not remember whether the path it took last time was longer or shorter.

Higher organisms, however, appear to operate on more robust representations of the world than the ant. If a chimpanzee is carried to a location where food is hidden (in an area it is familiar with), it will take the most direct route back to the food regardless of the path it was originally shown by the human. This requires a significantly more complex representation than the ant's, one that is more permanent and can be manipulated to abstract new information.

The environment of the Creatures raises a further issue. Before work began on the Navlab project, Chuck Thorpe and his colleagues had been developing mobile robots functionally similar to Brooks's. These early robots were completely autonomous and could make their way around buildings, labs, and classrooms. When the Navlab project began in 1984, the goal was to develop a system that could function outdoors. To the researchers' chagrin, the indoor robots failed badly when transplanted to natural environments. The Navlab required a major rethinking of all aspects of the task, and especially of the functional capabilities it required. It is consequently unclear whether Brooks and his Creatures are on the right track towards fully autonomous systems that can function in a wider variety of environments.

Pengi

Philip Agre and David Chapman (1987) wrote a program called Pengi that plays Pengo, an arcade-style video game. Pengo's rules are simple, but it makes severe real-time demands on human players. The authors' goal was to develop a model of activity in this environment that requires no explicit planning or a representation of the environment. Following Suchman (1987), the authors argued that planning plays an insignificant role in our everyday interaction with the world. "Before and beneath any activity of plan following, life is a continual improvisation, a matter of deciding what to do now based on how the world is now" (Agre & Chapman, 1987, p. 268).

Pengo is a two-dimensional maze where the agent manipulated by the player is a penguin that is chased by bees. The maze is built of ice blocks that the penguin can kick in order to kill bees. The structure of the maze is modified as ice blocks are kicked about. Pengi's knowledge of the world consists of a set of rules (e.g., "When you are being chased, run away") and a set of indexical-functional representations of the objects on the screen (e.g., "the-block-I'm-pushing" or "the-corridor-I-am-running-along").

The rules act as routines (which Agre & Chapman, 1987, stressed should not be confused with plans) by combining to form patterns of activity. Rules are sensitive to changes on the screen in the sense that if the conditions that support them are removed (e.g., a bee stops chasing the penguin), they are no longer applied. Some other rule with applicable conditions then takes over. Pengi therefore requires, a priori, a set of rules that cover every possible state of the game.

Agre and Chapman (1987) stated that there are three central characteristics of interaction with the real world. First, there is real-time involvement: It is a requirement of any intelligent agent that it be able to respond immediately to environmental stimuli. Second, the real world is largely uncertain: The behavior of the objects to be interacted with cannot be fully predicted. Third, the real world is complex: They define this characteristic only in terms of the combinatorial explosion produced by simple look-ahead search planning. More explicitly, what makes a task more or less complex is the contingency of behavior on what else is going on in real time.

> Naturally we ascribe the player's seeming purposefulness to its models of its environment, its reasoning about the world, and its planful efforts to carry out its tasks. But as with Simon's ant the complexity of the players' activity may be the result of the interaction of simple opportunistic strategies with a complex world. (Agre & Chapman, 1987, p. 269)

Pengi and the ant have much in common. A genuinely complex task requires perceptually and temporally unavailable elements to be organized into a coherent pattern of activity. Complexity increases if this organization is carried out in real time while interacting with a constantly changing environment. However, a Pengo player manipulates only one agent, the penguin, and thus does not need to integrate its activity with that of any other agents in order to achieve a goal. Furthermore, all the objects in the Pengo world are visually available to Pengi. It would seem, therefore, that the Pengo task does not meet either criterion for complexity. (Chapman & Agre granted that some tasks do require planning, although they do not attribute this requirement to increased complexity.)

A general misapprehension held by SA researchers about planning systems was stated quite clearly by Chapman (1989) "Planners are designed to solve problems; technically defined, a problem is a sort of logical puzzle that can be solved once and for all" (p. 49). Because a video game is an ongoing activity, it cannot be solved once and for all, and therefore it is not a problem in the traditional sense. Pushing Chapman's statement one step further, if it is not a problem, then traditional planners cannot be applied to it. This view of planners and problems seems very outdated when one considers projects like Phoenix and Navlab where flexible plans are able to address ongoing problems in real time.

Chapman (1989) also created a system that solves a seemingly more complex problem. The fruitcake problem, as it is called, is much like most blocks-world problems except that the blocks have letters on them; the goal is to create a stack that spells a particular word ("fruitcake" in this case).

Chapman's motivation for adapting Pengi to this task was to demonstrate that such a problem could be solved using a complex visual system and no representation. "Representation is generally thought to make problems easy; if a problem seems hard, you probably need more representation. In concrete activity, however, representation mostly just gets in the way" (p. 48).

This system (which Chapman, 1989, called Blockhead) uses markers to determine the spatial relations among blocks. For example, to determine whether a block to be moved has other blocks on top of it, one marker is placed on the desired block and another one moves up the stack until it is not over any block (i.e., it's at the top of the stack). The top marker is moved down one unit to mark the block at the top of the stack. If the two markers are seen to be next to one another, then this means that there are no blocks on top of the desired block and it can be moved. Otherwise, the block at the top of the stack is removed and the procedure is carried out recursively until the desired block has no blocks on top of it.

Chapman (1989) and Agre and Chapman (1987) argued that Pengi and Blockhead do not use symbols in the traditional sense because they do not have variables that they bind to constants in the world. "The-bee-I'm-chasing" could refer to any bee on the screen at different points in time. Also the very same bee could turn into "the-bee-that-is-chasing-me." In the case of Blockhead, two coincident markers means that the top of the block is clear, no matter what block it is. If we adopt this logic, then ordinary language, which uses such phrases, and also simpler ones like "that bee" or "that block," or still simpler pronouns like "this" and "she" must be regarded as nonsymbolic, a surprising outcome.

The claim that these systems do not use representations clearly rests on an unusual definition of the term symbol (certainly not the one given at the outset of this article). Both Brooks's (1991) insects and Agre and Chapman's (1987) Pengi seem to have categorical representations of states in the world and functional characterizations of those states. With these characteristics, they satisfy the definition of a symbol system. That the symbols in question are both goal-dependent and situation-dependent does not change their status. They are genuine symbols in the traditional information-processing sense.

In particular, it seems that Agre and Chapman (1987) based their claim on a misunderstanding of the fact that some symbols represent a distinct thing or state in the world. This fact does not entail that there needs to be a fixed object bound to a symbolic variable. "The-bee-that-is-chasing-me" is a perfectly good symbol; it denotes a distinct class of objects in the world (i.e., any bee that is engaging in the activity of chasing me). It is no different in kind from "the-largest-disk-not-on-the-goal-peg" or "the-largest-blocking-disk" symbols used in the perceptual program for solving the Tower of Hanoi puzzle.

It is not necessary to reach for a new theoretical paradigm in order to create an implementation that can play a video game. John, Vera, and Newell (in press) implemented a model of an expert Super Mario Bros. 3® player. This model is based on Soar, a paradigmatic example of symbol system architecture. Like Agre and Chapman's (1987), this model also uses indexical-functional representations of objects on the screen. There are attributes, such as "closest-enemy," that can have different objects as values. As with Pengi, the true challenge in developing this model has been learning, again suggesting that the biggest difficulty in creating such models lies not on the side of immediate behavior, but on the side of reorganizing the system's information into more useful and permanent forms by learning.

4. THEORETICAL ISSUES

The proponents of SA, in contrasting it with the established symbolic viewpoint in cognitive science, have at one time or another made the following claims:

1. SA, unlike symbolic computation, requires no internal representations.
2. SA operates directly between the agent and the environment without the mediation of internal plans.
3. SA postulates direct access to the "affordances" of the environment. That is, the actor deals with the environment, and with his or her own actions, in purely functional terms.
4. SA does not use productions.
5. The environment, for purposes of SA, is defined socially, not individually or in physicalist terms.
6. SA makes no use of symbols.

On the basis of the discussion in previous sections of this article, we are now in a position to take up these claims, examining the evidence that supports or refutes them.

Internal Representations

In some situations, an actor's internal representations can be extremely simple, but no one has described a system capable of intelligent action that does not employ at least rudimentary representations. Perhaps the barest representation encompasses only goals and some symbolization of a relation between goal and situation, on the one hand, and action on the other. But some internal representation of these is unavoidable if action is to be purposive.

At a minimum, the relation between situation and action can be represented by a single production, the condition of which is a symbol denoting the situation, the action a symbol denoting the response. These symbols can be regarded as denoting functions (or, more precisely, as denoting the recognition and response sides of the same function, e.g., "wash," *n.* and "wash," *v.*).[5]

In the case of breakdown, where action does not accomplish the goal "effortlessly," the information about the situation must be elaborated at least down to the level where diagnostic cues and repair activities can be represented. The information for the elaboration can come both through encoding additional externally perceived stimuli, and through access to information already stored in memory by previous experience and learning.

Viewed from the level next above, each level of elaboration looks like a description of a mechanism or system of mechanisms for accomplishing the function. Viewed from the level next below, each level of aggregation looks like the name of a function performed by the mechanism. Functions can be distinguished from mechanisms only relatively. All depends on the vantage point from which they are viewed.

There is abundant empirical evidence that human agents use elaborate problem representations in difficult situations. Many of these representations, at least for many people in many situations, have a pictorial, diagrammatic, or "imaginal" character. We need not settle here the precise form of a "mental picture." The behavioral manifestation that the actor has one is that he or she reports "seeing" the situation or some aspect of it in the "mind's eye," and is able to extract from the image various kinds of information about it in a way similar to the way in which information would be extracted from an external visual display.

Using symbolic means, internal images can be constructed in computer memories that have many of the properties human subjects report about their mental images. Such images require procedures for encoding objects and their spatial relations, and encoding processes that can extract information from the images or act to modify them.

In systems like Pengi (Agre & Chapman, 1987) and the creatures of Brooks (1991), often taken as paradigmatic examples of applied SA, there are substantial internal representations, some of them used to symbolize the current focus of attention and the locations of relevant nearby objects, others used to characterize the objects themselves in terms of their current functions. If particular locations in the external representation are "marked," as in the application of Pengi to the Fruitcake problem, the markers can only be interpreted as elements of the internal representation (e.g., as objects of a

[5] Notice that SA here bears a striking resemblance to classical S→R behaviorism, rather ironic in view of the distaste of most proponents of SA for "mechanistic" explanation, and their preference for holistic accounts of perception. Of course (and B.F. Skinner was always at pains to point this out), the "S" in such behaviorism includes not only the (interpreted) stimulus of the moment, but the whole previous history of stimuli that have left residuals capable of affecting behavior in memory.

focus of attention): They are not physically attached to the external objects or locations, and if they were so attached (by a paint sprayer?), they would have to be reinterpreted internally each time they were sought out and detected.

Observing that an agent like Pengi maintains certain causal relations between itself and the world does not explain *how* such relations are maintained. It is quite reasonable to suggest that an agent maintains an orientation toward an object through a causal relation with it, and that this relation is best construed as a *pattern of interaction*. However, it is unreasonable to suggest that a pattern of interaction is produced magically without any corresponding change in the representational state of the agent. There is no reason to believe that it can be produced without at least a minimal representation (such as the markers that Pengi uses). To be sure, Pengi is only a first attempt; but there is no evidence that subsequent attempts will solve the problem without interpretable perceptual markers of some kind.

In sum, there is no evidence for cognition without at least minimal representation, and when there is anything problematic about response, the internal representations used to generate the response may be elaborate indeed, and in particular, may incorporate mental imagery. These internal representations constitute an important part of the context, including social context, to which behavior is responsive. Moreover, the internal representations have all the properties of symbol structures.

Plans

Plans project potential action into a future time. Plans are almost always "abstract," in that they require lower level implementation for their execution. Typically, plans influence human action in two ways.

First, plans may be used to determine what initial (present) action will lead toward desired goals. For example, all of the look-ahead that a chess master carries out before making a move has, as its purpose, evaluating that move in comparison with other available moves. It is used to select only that next move, and does not commit the player to carrying out any of the subsequent steps of the plan that are foreseen. A new analysis, of greater or lesser extent, will be made after the environment (the opponent) has responded.

Second, plans may be used to establish a set of "islands," subgoals along the route to some distant goal. The use of an abstract planning space to fix such subgoals can reduce enormously the requirements for computations in a complex situation. Again, whether the plan will be followed or not will depend on sensory and perceptual feedback of information from intervening events. Plans in which successive actions are made conditional on information about the current situation are called strategies. Most of the plans that people use are probably strategies in this sense.

The shooting-the-rapids example illustrates one important way in which experts make use of plans. The plan normally consists of an approximate path through the rapids, as broad as feasible, taking advantage of the main currents and avoiding the obvious obstacles. In execution, avoidance of immediate obstacles will take precedence over the plan; but when such short-term problems have been met, and if new ones have not meanwhile arisen, the expert canoers will fix upon the subgoal of returning to the long-term path sketched out by the plan. In fact, the most important property of such a plan is that it minimizes the number of occasions when an emergency calling for situated action will arise. ("Driving defensively" is a well-known form of planning with similar properties.)

There is then no contradiction between the view that human beings form plans and that their behavior is influenced by them, and the view that much action, in the face of severe real-time requirements, is situated action based on rather meager representations of the situation. Both aspects of behavior are observable, commonly in the same humans within the same complex activity. Both forms of action require some internal representation of the situation—perhaps minimal in the case of situated action, more elaborate in the case of planned behavior when fewer unexpected events occur.

Affordances

We have already seen that when people are dealing with familiar situations, using habitual actions, their internal representations, at the conscious level, may be almost wholly functional, without any details of the mechanisms that carry out these functions. The "affordances" of the environment, represented internally, trigger actions.

Of course, the absence of consciousness of mechanisms implies neither that mechanisms are absent nor that they are nonsymbolic. To acquire an internal representation of an affordance, a person must carry out a complex encoding of the sensory stimuli that impinge on eye and ear. And to take the corresponding action, he or she must decode the encoded symbol representing the action into signals to the muscles.

Ironically, affordances, far from removing the need for internal representations, *are* carefully and simply encoded internal representations of complex configurations of external objects, the encodings capturing the functional significance of the objects. Affordances, more familiarly known as "chunks," have long played an important role in symbolic information-processing systems that solve problems, learn, and perceive. EPAM and Soar are well-known systems of this kind, the former of which, at least, has shown its ability to account in considerable detail for a wide range of psychological phenomena in learning, memory, and perceptual tasks.

Productions

Connections between stimuli and responses may be genetic or they may be learned, or both. The simplest way to represent them is as productions, which at the highest (functional) level can be symbolized as: C→ A. The condition of a production may then be expanded into a more or less simple or complex set of tests (on both sensory and/or stored information), and the action into a more or less simple or complex set of messages through the efferent nervous system. The sensory tests provide the external context for the action, the tests on memory contents provide the internal context, including the goal.

If a condition-action connection is "wired in," we do not need to think of it as symbolized. But even if it is not so regarded, relations of denotation are present. On the condition end, the neural impulse aroused by the encoded incoming stimuli denotes the affordances that produced these stimuli, while the signals to efferent nerves denote the functions of the actions. There is every reason to regard these impulses and signals as symbols: A symbol can as readily consist of the activation of a neuron as it can of the creation of a tiny magnetic field.

Hence, the use of productions to implement the internal processes of thought makes no commitment to the precise way in which neurons operate, or to the exact dynamic structures in the brain that are to be regarded as symbols. Productions provide an essentially neutral language for describing the linkages between information and action at any desired (sufficiently high) level of aggregation. If we are concerned with events taking several hundred milliseconds, it is probably almost essential to regard productions as operating on symbols. If we are concerned with events lasting a few milliseconds, we may wish to use a language of neurons rather than symbols.

The Social Environment

All human behavior is social. First and foremost, it is social because almost all the contents of memory, which provide half of the context of behavior, are acquired through social processes—processes of learning through instruction and social interaction. Not only is memory acquired through social processes, but a large part, perhaps the major part, is social in content: information about specific people, or about people in general and their modes of interaction.

Behavior is social also because the other half of its context is provided by an environment that, on most occasions, is highly social, too. Not only the contexts of conversation and interpersonal interaction, but also the contexts of reading and interacting with social artifacts (plows, computers, hunting spears), are all thoroughly social.

For many purposes of cognitive simulation, it is of no special significance that thought is social. So long as a system is provided with a knowledge base that corresponds with the relevant knowledge possessed by the person who is being simulated, one need not be concerned with the original source of that knowledge. A theory of performance (e.g., problem-solving performance) explains how performance processes gradually transform given initial conditions into new knowledge that includes a path to the goal.

A quite different task of explanation is to show how, and in what form, the body of knowledge comes to be stored in memory: how it is learned. Learning can be studied primarily in terms of its internal mechanisms, again taking the input (e.g., material from a textbook) as given, and seeking to model how that input changes the internal contents of memory so that the system will subsequently possess the desired skill or the desired knowledge.

On the other hand, the learning process can be connected with its social surround by extending the inquiry to explaining how just that textbook was used or produced, or how particular knowledge in the textbook came into being. If the inquiry is extended to this last question, it becomes an investigation of the processes of scientific discovery, requiring its own models for simulation.

The topic of scientific inquiry is an interesting one for examining the relations of the individual and the social. Most simulation of scientific discovery has been carried out as if the successive steps of discovery were the work of a single scientist. So, for example, the program KEKADA was used to simulate several months of experimentation of the biochemist Krebs (actually, Krebs and his doctoral student).

Only passive elements in KEKADA's simulation were explicitly social: The biochemical and biological knowledge with which Krebs began his experimentation, and his later probes into the literature to check a couple of conjectures. Of course Krebs's own processes of inquiry were also formed in a social environment, in particular, the environment of Otto Warburg's laboratory. Hence, if we did not wish to take the experimental strategies as given, we would have to study the learning processes of a graduate or postdoctoral student in such a laboratory. Tracing matters even further back, we could try to simulate how Warburg invented his experimental strategy, for example, his method of tissue slices.

A simulation program need not be limited to the work of a single scientist: It can encompass a research program of a whole group of scientists, either dealing with the mutual transfer of knowledge among them (a complex program), or simply aggregating the system to treat their collective knowledge as a shared memory.

The preceding paragraphs suggest, however sketchily, how social processes can be brought into a cognitive simulation. There is no need to simulate the whole world in a single model. Sequences of events (perhaps involving a single scientist) can be studied in isolation, the initial state of knowledge and the communications received in the course of the process being treated as exogenous variables. Another simulation can, in turn, be aimed at explaining some of these same variables and the processes that determined their values.

Different component models can explain phenomena at different levels of aggregation. An example of simulation of a rather aggregated social process can be found in Chapter 7 of *Scientific discovery* (Langley et al., 1987), which describes the development of theories of combustion from the early phlogiston theory to the theory of oxidation. It was a process extending over a generation, in which many scientists participated.

Symbols

Our examination of SA has unearthed no reasons why mental processes cannot be represented as the processes of a physical system. There may be some question as to whether this is always the most convenient representation, but its feasibility has been placed beyond question by innumerable examples, over the past 35 years, of its successful use. A number of these examples have been referred to in this article, and a number of conjectured counterexamples have been refuted.

In much of the SA literature discussed here, there appears to be confusion between the question of whether certain mental events are symbolic in character and the question of whether these events are within the conscious awareness of the actor. There is no essential connection between symbols and consciousness. The information in DNA and RNA is certainly represented symbolically, the symbols having clear denotations, but this information is not in the organism's conscious awareness.

In fact, most simulations in the literature of information-processing psychology do not draw a clear boundary between what information is available to the consciousness of the actor, and what information is not (an ambiguity that is important for some applications, but not for others). The nearest approach to such a boundary is found in programs that provide a clear separation between information in short-term memory and information in long-term memory. The actor is presumably aware of the former, but not of the latter.

Simulations that use spreading-activation mechanisms may be interpreted to imply that information in the currently activated portion of memory is available to consciousness, whereas other information in memory is not. We are not evaluating here the psychological accuracy of any of these theories, but simply warning against confusing symbolic representation with awareness.

It is well known from empirical evidence that people are consciously aware of a person or thing they have just recognized, but not aware of the processes that led from perceptual cues to the recognition. The EPAM program, a simulation of elementary perceptual and memorizing processes, represents in some detail the process of recognition, a wholly unconscious process. The program accomplishes this by sorting through a discrimination net the encoded (and symbolic) representation of the perceptual information. To do this, it applies a succession of tests to this symbol structure.

Thus EPAM simulates both symbols that are available to consciousness (the symbols denoting recognized objects and providing access to information about them stored in LTM), and symbols of which the actor is unaware (the encoded perceptual symbol structures that are sorted through the discrimination net).

Context

Finally, we return to the central claim of hard SA: that behavior can only be understood in the context of complex real-world situations. Interpreted literally, this claim is surely wrong, because no organism, natural or artificial, ever deals with the real-world situation in its full complexity.

An SA system extracts "affordances" (symbols denoting functions) from the environment and responds to them. It survives if the environment is sufficiently subdivided into semi-independent components and sufficiently empty that this strategy does not ignore relevant environmental circumstances that have to be responded to in real time. Complexity is handled largely by the strategy for focus of attention (hence, priority of processing), by sophisticated, learned schemes for encoding environmental patterns as affordances (e.g., "attacking bee") and by sophisticated strategies for responding to them. Previously stored knowledge about the exact state of the environment plays only a small role in such systems, with the important proviso that if the environment stages "surprises" that go beyond the available affordances, the system is likely to ignore them or behave inappropriately.

Planning systems undertake to supplement the predetermined responses of pure SA systems with capabilities for building (very simplified) models of the real world, and using these models to plan actions and predict real-world responses. If the world does not behave exactly as the simplified model predicts, such systems will not long survive unless they also have good capabilities for detecting and responding to these deviations (i.e., situated action capabilities). These reactions to information from the environment allow the system, using as its strategies, both to react rapidly to the real situation and to modify its internal model for subsequent planning.

In the light of these considerations, we think a defensible claim, to replace the invalid one of hard SA, is that behavior can only be understood in the context of environments that change continually, and whose complexity is so great that only extremely simplified approximations of them can be handled by the systems's response mechanisms or its planning mechanisms, severally or jointly. Bounded rationality is the name of the game, and it is as surely present in a game of chess as in any of the games that humans play in what they call "the real world."

Of course, this claim does not address the pedagogical question of whether, for purposes of learning, it is best to expose humans to real-world situations or to situations of the kinds that are more traditional in the classroom. The correct answer is probably "sometimes," but establishing that answer, and the conditions that define it, is a task for future research.

5. CONCLUSION

We have examined the claims of SA without finding reasons why such action cannot be accommodated within the physical symbol-system hypothesis. The hypothesis asserts that intelligent behavior is the product of systems that can handle patterns of arbitrary variety and complexity; that can construct complex structures of such patterns, and store and modify such structures in memory; that can input such patterns through the encoding of sensory information, and output them through the innervation of motor neurons; and that can compare patterns, behaving one way if the patterns match, another way if they do not. It is the ability to perform these functions, the functions of a physical symbol system, that provides the necessary and sufficient condition for behaving intelligently: responsively to the needs and goals of the organism and to the requirements imposed on it by the environment.

It follows that there is no need, contrary to what followers of SA seem sometimes to claim, for cognitive psychology to adopt a whole new language and research agenda, breaking completely from traditional (symbolic) cognitive theories. SA is not a new approach to cognition, much less a new school of cognitive psychology. Whether particular forms of human behavior meet the criteria of SA is an empirical question whose answer is certainly different for different behaviors. But whatever the answer, complex human behavior, whether it has been labeled "situated" or not, can be and has been described and simulated effectively in physical symbol systems.

Early AI did not pay much attention to the distinction between internal and external representations. In a program for solving the Tower of Hanoi puzzle, the representation of the disks and pegs can be considered to be either a mental representation, held in the actor's memory, or a representation of the real world of disks and pegs "out there." Earlier, we showed how the program could be solved in either representation by symbolic programs.

Research in robotics has had the valuable effect of calling attention to important properties of external representations. External real-world situations are far too rich and complex to be captured fully and accurately by a robot's internal models of them. Consequently, there must be continual feedback to test the actions proposed by the internal representation against reality, and to correct that representation to reflect where the robot really is, and what the external world is really like.

If action is to be taken in real time, the internal representation must be kept simple, stripped down to the functional essentials. This implies that there must be a sophisticated functional encoding of the affordances of the external situation, and a corresponding decoding of functionally described actions into motor sequences. In living organisms, these encodings and decodings are facilitated by special-purpose perceptual and motor processes that have evolved over millennia and eons.

The term "situated action" can best serve as a name for those symbolic systems that are specifically designated to operate adaptively in real time in complex environments. SA, so interpreted, has played and will continue to play an important role in the development of robotics, and in cognitive theories of human (and other animal) interactions with the environment. It will provide an essential component of the theory of physical symbol systems. It in no sense implies a repudiation of the hypothesis that intelligence is fundamentally a property of appropriately programmed symbol systems.

REFERENCES

Agre, P.E., & Chapman, D. (1987). Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence,* 268–272. Menlo Park, CA: American Association for Artificial Intelligence.

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

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

Atkinson, R.C., & Schiffrin, R.M. (1968). Human memory: A proposed system and its control process. *The Psychology of Learning and Motivation, 2,* 89–195.

Barsalou, L.W. (1987). The instability of graded structure: Implications for the nature of concepts. In U. Neisser (Ed.), *Concepts and conceptual development: Ecological and intellectual factors in categorization.* Cambridge, MA: Cambridge University Press.

Brooks, R. (1986). A robust-layered control system for a mobile robot. *IEEE Journal of Robotics and Automation, RA-2,* 14–23.

Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence, 47,* 139–159.

Card, S., Moran, T.P., & Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, NJ: Erlbaum.

Chapman, D. (1989). Penguins can make cake. *AI Magazine, 10,* 45–50.

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.

De la Rocha, O. (1984). The dialectic of arithmetic in grocery shopping. In B. Rogoff & J. Lave (Eds.), *Everyday cognition: Its development in social context.* Cambridge, MA: Harvard University Press.

Ericsson, K.A., & Simon, H.A. (1984). *Protocol analysis.* Cambridge, MA: MIT Press.

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

Greeno, J.G. (1989). Situations, mental models and generative knowledge. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon.* Hillsdale, NJ: Erlbaum.

John, B.E., Vera, A.H., & Newell, A. (in press). Toward real-time GOMS: A model of expert behavior in a highly interactive task. *Behaviour and Information Technology.*

Kaplan, C., & Simon, H.S. (1990). In search of insight. *Cognitive Psychology, 22,* 374-419.

Kripke, S. (1972). Naming and necessity. In D. Davidson & G. Harman (Eds.), *Semantics of natural language.* Dordrecht: D. Reidel.

Langley, P., Simon, H.A., Bradshaw, G., & Zytkow, J. (1987). *Scientific discovery.* Cambridge, MA: MIT Press.

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

Nilsson, N.J. (1988). *Memorandum.* Stanford University.

Putnam, H. (1975). The meaning of meaning. In Gunderson (Ed.), *Language, mind, and knowledge.* Minneapolis: University of Minnesota Press.

Pomerleau, D.A., Gowdy, J., & Thorpe, C.E. (1991). Combining artificial neural networks and symbolic processing for autonomous robot guidance. *Engineering Applications of Artificial Intelligence, 4,* 961-967.

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

Simon, H.A. (1973). The structure of ill-structured problems. *Artificial Intelligence, 4,* 181-201.

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

Simon, H.A. (1989). *Models of thought* (vol. 2). New Haven, CT: Yale University Press.

Suchman, L.A. (1987). *Plans and situated action: The problem of human-machine communication.* New York: Cambridge University Press.

Thorpe, C.E. (1990). *Vision and navigation: The Carnegie Mellon Navlab.* Norwell, MA: Kluwer.

Winograd, T., & Flores, F. (1986). *Understanding computers and cognition: A new foundation for design.* Norwood, NJ: Ablex.