Unified Theories of Cognition

Allen Newell

Harvard University Press
Cambridge, Massachusetts
London, England
1990

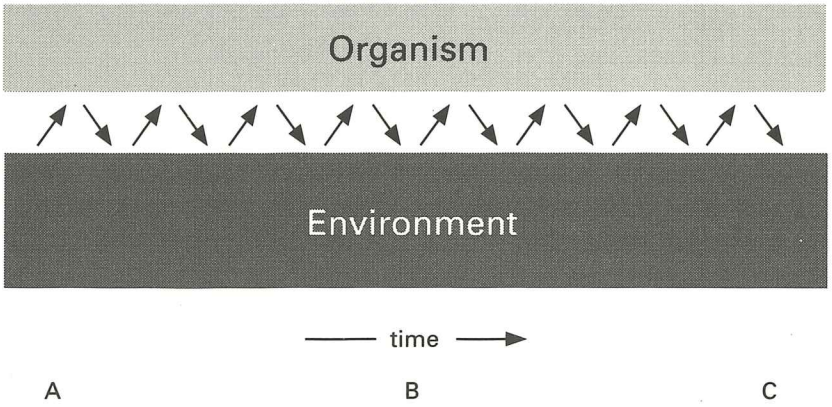

Figure 2-1. Abstract view of mind as a controller of a behaving system.

other yet something entirely different. When you come back to the same situation you may have a different response function. You climb into the car and something happened differently and you can't remember where the key is and now you do something different. The world is divided up into microepics, which are sufficiently distinct and independent so that the control system (that is, the mind) produces different response functions, one after the other.

It is certainly possible to step back and treat the mind as one big monster response function; that is, treat the mind as a single function from the total environment over the total past of the organism to future actions (under the constraint that output at a given time never depends on future input). Describing behavior as multiple response functions implies some sort of decomposition within the organism. In effect the organism treats the environment as different enough from time to time, so that the aspects that enter into the function (that the behavior is made a function of) have little in common. Thus it is possible to describe the organism as using separate functions, one for each situation. This phrase, *response function,* will occur over and over again throughout the book, so remember it.

2.2. Knowledge Systems

How then should we describe systems? How should we describe their response functions? To speak of mind as a controller suggests

immediately the language of control systems—of feedback, gain, oscillation, damping, and so on. It is a language that allows us to describe systems as *purposive* (Rosenbleuth, Weiner, & Bigelow, 1943). But we are interested in the full range of human behavior and response—not only walking down a road or tracking a flying bird, but reading bird books, planning the walk, taking instructions to get to the place, identifying distinct species, counting the new additions to the life list of birds seen, and holding conversations about it all afterward. When the scope of behavior extends this broadly, it becomes evident that the language of control systems is really locked to a specific environment and class of tasks—to continuous motor movement with the aim of pointing or following. For the rest it becomes metaphorical.

A way to describe the behavior of systems with wide-ranging capability is in terms of their having *knowledge* and behaving in light of it. Let us first see what that means, before we see how we do it. Figure 2-2 shows a simple situation, the *blocks world,* which is suitably paradigmatic for systems characterized as having knowledge. There is a table, on which sit three blocks, *A*, *B*, and *C*, with block *A* on top of block *B*. Some agent *X* observes the situation, so that we can say that *X* knows that *A* is on top of *B*. Another agent, *Y*, who does not observe the situation, asks *X* whether *B* is clear on top. We say, almost without thinking, that *X* will tell *Y* that *B* is not clear. We have actually made a prediction of *X*'s response. Let us say that is exactly what happens (it is certainly plausible, is it not?). What is behind our being able to predict *X*'s behavior?

A straightforward analysis runs as follows. We assume that *X* has a goal to answer *Y* truthfully. There must be a goal involved. If we can't assume any goals for this agent, then no basis exists for predicting that *X* will answer the question, rather than walking out of the room or doing any other thing. The agent's goal (in this case) is something like this: if someone asks a simple question, answer truthfully. We take it that if *X* knows something, *X* can use that knowledge for whatever purposes it chooses. Thus, we calculate: *X* knows that block *A* is on top of block *B*; *X* wants to answer the question truthfully; *X* has the ability to answer (*X* can communicate); consequently, *X* will tell *Y* that block *B* is not clear on top. Thus, we can predict what *X* will do. The prediction need not always be right—we may be wrong about *X*'s goals, or about what

Let X observe a table of stacked blocks
We say "X knows that A is on top of B"
A nonobserver Y asks X whether B is clear on top
We say (predict) "X will tell Y that B is not clear"

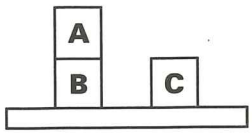

Figure 2-2. The simple blocks world.

X knows, or some other aspect of the situation that could prevent the action. Still, this is a useful scheme to predict a system's behavior.

The analysis of knowledge has a long philosophical history, and indeed constitutes the standard subarea of epistemology. It has a continuation within cognitive science (Goldman, 1986). That analysis takes knowledge as something sui generis—something special with special issues about what it means for a system to have knowledge and especially what it means for knowledge to be certain. What I claim cognitive science needs, instead, is a concept of knowledge that is used simply to describe and predict the response functions of a system. There can be, of course, a different response function for every goal and every body of knowledge, but the little situation with the blocks is entirely paradigmatic of our use of the concept. It is a way of characterizing a system such that we can predict (with varying degrees of success) the behavior of the system.

Thus, to treat something as having knowledge is to treat it as a system of a certain kind. We always describe systems in some way, if we are to deal with them at all. Some systems we describe in one way, some in another. Often we describe the same system in multiple ways. To describe a system as a *knowledge system* is just one of the alternatives that is available. The choice of what description to use is a pragmatic one, depending on our purposes and our own knowledge of the system and its character.[2]

2. The use of the phrase "our purposes and our own knowledge" in order to describe the nature of knowledge is benign and does not indicate any vicious circle. To discuss when an agent uses a given type of description, we must describe that agent. In this instance, the appropriate description for the agent, which is us, is as a knowledge system.

Knowledge-level systems
Medium: Knowledge
Laws: Principle of rationality

Program-level systems
Medium: Data structures, programs
Laws: Sequential interpretation of programs

Register-transfer systems
Medium: Bit vectors
Laws: Parallel logic

Logic circuits
Medium: Bits
Laws: Boolean algebra

Electrical circuits
Medium: Voltage/current
Laws: Ohm's law, Kirchhoff's law

Electronic devices
Medium: Electrons
Laws: Electron physics

Figure 2-3. The hierarchy of computer systems.

Consider the familiar computer-systems hierarchy, shown in Figure 2-3, which we will encounter repeatedly in the course of this book. A computer system can be described in many ways. It can be described as a system of electronic devices, or as an electrical circuit, or as a logic circuit, or as a register-transfer system, or as a programming system. There are other ways as well, ways that are not related to its primary function, such as an item of cost in a budget, a contributor to floor loading, or an emblem of being high-tech. All the descriptions based on the computer as a behaving system are types of *machines*. In each case there is some kind of *medium* that is processed. Working up from the bottom, the media are electrons, current, bits, bit vectors, and data structures. At any moment, the *state* of the system consists of some configuration of its medium. There are *behavior laws* that can be used to predict the behavior of the system. Electrons are particles that move under impressed electromagnetic forces; electrical circuits obey Ohm's law and Kirchhoff's law; logic circuits obey Boolean algebra; the processing in programs obeys the stipulated programming language. In each case, if we know the state of the system and the laws of its behavior, we can obtain the state of the system at some point

in the future. Each of these descriptions provides a different way to make predictions about system behavior.

A clarification is in order. All along, I keep referring to predictions. This is simply shorthand for all the various uses of descriptions, such as explaining behavior, controlling behavior, or constructing something that behaves to specifications. Although there are differences in these activities and some descriptions are more suited to one than the other, it becomes tiresome to have to be explicit each and every time. "Prediction" will cover them all.

The descriptions of computer systems form a hierarchy of *levels,* because each higher-level description is both an abstraction and a specialization of the one below it. Consider the level of electrical circuits. Its medium, current, is the flow of electrons. Its laws, Ohm's and Kirchhoff's, can be derived from electromagnetic theory, specialized to networks of conductors. Or consider the program level. Data structures are sets of bit vectors, to be interpreted in fixed ways by various operations. The operations can be described as the outcomes of specific register-transfer systems, as can the interpretation of a program data structure that determines which operations are executed. Each level abstracts from many details of the level below.

Systems become more specialized as the hierarchy is ascended. Not every system describable as an electrical circuit is also describable as a logic circuit. Not every system describable as a register-transfer system is a programmable system. The relationships between the levels are sometimes quite transparent, as is the simple aggregation that connects the logic level to the register-transfer level, where bits are simply organized into vectors of fixed length and handled in a uniform way, except for a few special operations (such as addition and multiplication, with their carries). Sometimes the relationships are less obvious. Inventing electrical circuits that behaved discretely according to the laws of Boolean logic required a rather substantial evolution, mediated by the work on pulse systems for radar.

Knowledge systems are just another level within this same hierarchy, another way to describe a system. As a level in the hierarchy, knowledge is above the program level in Figure 2-3. The knowledge level abstracts completely from the internal processing and the internal representation. Thus, all that is left is the content of the representations and the goals toward which that content will be

used. As a level, it has a medium, namely, knowledge. It has a law of behavior, namely, if the system wants to attain goal *G* and knows that to do act *A* will lead to attaining *G*, then it will do *A*. This law is a simple form of rationality—that an agent will operate in its own best interests according to what it knows.

As just another level in the hierarchy of Figure 2-3, there is nothing special about the knowledge level, in any foundational or philosophical sense. Of course, the knowledge level is certainly different from all the other levels. It has its own medium and its own laws and these have their own peculiarities. But, equally, each of the other levels is different from all the others, each with its own peculiarities. The levels can, of course, also be classified in various ways, such as discrete versus continuous, or sequential versus parallel. But the classification is not very important, compared with the individual particularity of each level, in how it describes a system and what are the characteristic modes of analysis and synthesis that go with it so that it can be used effectively.

Descriptive schemes like the one put forth in Figure 2-3 do not carry with them obvious scientific claims. They seem to be simply ways of describing parts of nature. However, they are far from theoretically neutral. Scientific claims arise when we discover (or assert) that such a descriptive scheme can actually be used successfully, or with such and such a degree of approximation, for a given real system or type of system. The criterion for success is that the system is operationally complete—its behavior is determined by the behavior laws, as formulated for that level, applying to its state, as described at that level. The claim is that abstraction to the particular level involved still preserves all that is relevant for future behavior described at that level of abstraction. The force of such claims can be appreciated easily enough by imagining someone handing you a small closed box and asserting, "There is a programmable computer inside." This means you will find something inside that can be successfully described as a programmable computer, so that you may treat it so, expecting to be able to program it, execute it with a loaded program, and so on. Taking the abstraction as given and acting on your expectations, you would be in for one successful prediction after another (or failure thereof) about a region of the world you hitherto had not known.

Thus, to claim that humans can be described at the knowledge level is to claim there is a way of formulating them as agents that

have knowledge and goals, such that their behavior is successfully predicted by the law that says: all the person's knowledge is always used to attain the goals of the person. The claim, of course, need not be for completely successful prediction, but only to some approximation.

It is easy to see why describing a system at the knowledge level is useful. The essential feature is that no details of the actual internal processing are required. The behavior of an existing system can be calculated if you know the system's goals and what the system knows about its environment. Both can often be determined by direct observation—of the environment, on the one hand, and of the system's prior behavior, on the other. The knowledge level is also useful for designing systems whose internal workings are yet to be determined. The knowledge level provides a way of stating something about the desired behavior of the system and about what it must incorporate (namely, the requisite knowledge and goals). Specifications for systems are often given at the knowledge level. Every level, of course, can and does serve as a specification for the level below it. The special feature of the knowledge level is that it can be given before anything about the internal workings of the system is determined.

Let us summarize by restating rather carefully what a knowledge-level system is (Figure 2-4). A knowledge system is embedded in an external environment, with which it interacts by a set of possible actions. The behavior of the system is the sequence of actions taken in the environment over time. The system has goals about how the environment should be. Internally, the system processes a medium, called knowledge. Its body of knowledge is about its environment, its goals, its actions, and the relations between them. It has a single law of behavior: the system takes actions to attain its goals, using all the knowledge that it has. This law describes the results of how the knowledge is processed. The system can obtain new knowledge from external knowledge sources via some of its actions (which can be called perceptual actions). Once knowledge is acquired it is available forever after. The system is a single homogeneous body of knowledge, all of which is brought to bear on the determination of its actions. There is no loss of knowledge over time, though of course knowledge can be communicated to other systems.

Characterizing knowledge as the medium of a system level, which is just one system level among many, constitutes a particular

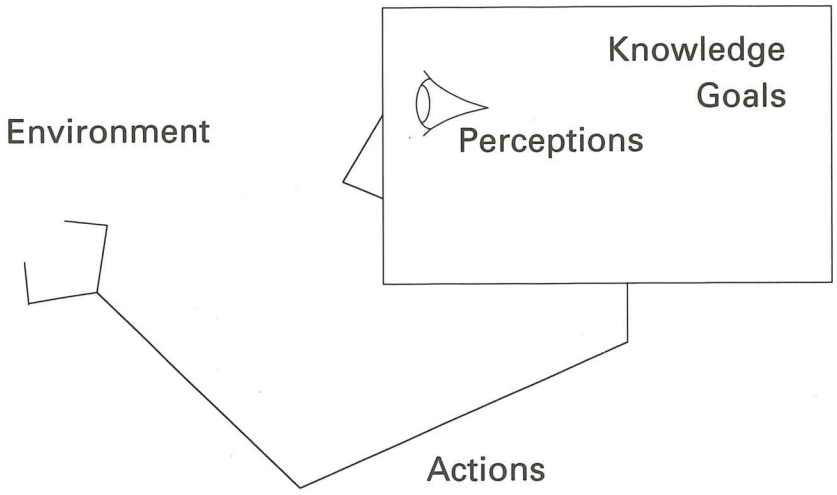

Figure 2-4. The knowledge-level system.

theory about the nature of knowledge. The existing extensive philosophic literature about knowledge does not describe knowledge in these terms. In general it does not describe knowledge in terms of a system at all, but simply proceeds to inquire after its validity and certainty. Daniel Dennett's (1978, 1988a) notion of an *intentional system,* however, is substantially the same as a knowledge-level system. Actually, the key concept for Dennett is that of the *intentional stance,* which is the way the observer chooses to view or conceptualize the agent.[3]

Although this knowledge-level systems theory is indeed a theory of knowledge, it is not in fact anybody's theory. It certainly is not *my* theory. I am not putting forth something that I have discovered or invented. Rather, this way of using knowledge systems is the actual standard practice in computer science and artificial intelligence. All that I have done is to observe the way we use this concept of knowledge and make it explicit.

It is useful to note that this theory of knowledge arose without specific authorship—without a specific inventor or discoverer. It

3. Dennett thus puts the emphasis on the nature of the observer rather than on the nature of what is observed. The reader interested in following up the way philosophers treat these matters and the relation of the intentional stance to the knowledge level can consult Dennett (1988b) and Newell (1988).

will help to sort out the various notions of knowledge that are around. The sociological structure of science, and scholarship more broadly, incorporates a view that ideas are authored by individuals of note, who are thereby to be honored and rewarded for producing and disseminating these ideas. Whatever view might be held about the ideas themselves, whether actually invented or merely discovered, they do not become part of science and society without the agency of particular men and women. There may be difficulties of determining who first discovered this or that idea. There may be genuine cases of simultaneous discovery (Merton, 1973), but some specific set of scientists or scholars still gets the credit. The case in hand doesn't fit this frame, however, nor do others coming later in the chapter, namely, symbols and architecture.

Computer scientists and engineers *as a group* developed what I argue is the appropriate theory of knowledge. They did so without any particular author laying out the theory. Lots of words, certainly, were written about computers and how to deal with them—from highly technical and creative efforts to general musings and on to popularizations and advertising copy. And some of these words have put forth novel ideas that can be said to have been authored, in perfect accordance with the standard view of science. John von Neumann is generally credited with *the stored-program concept* (though there is a modicum of dispute about it because of the events surrounding Eniac, Eckert and Mauchly, and the Moore School). But the stored-program concept (or any of the other ideas that were articulated) is not the notion of knowledge-level systems.

I do not know of any clear articulation of the idea of the knowledge level in computer science prior to my 1980 AAAI presidential address (Newell, 1982).[4] But that was almost twenty years after its use was common—after computer scientists were talking technically and usefully about what their programs knew and what they should know. All my paper did was give voice to the practice (and it was so represented in the paper). I have been, of course, a participant in the developing use of this notion, having been involved in both computer science and AI since the mid-1950s. And I have certainly done my share of writing scientific articles, putting forth

4. Dennett's writings on the intentional stance go back to the late 1960s but do not seem to owe much to computer science, on this score at least; see references in Dennett (1988a).

theories and concepts. But I was simply part of the community in how I learned to use such notions as knowledge. Here is a sentence and its explanatory footnote taken from an early paper (Newell, 1962, p. 403):

> For anything to happen in a machine some process must know* enough to make it happen.
>
> *We talk about routines "knowing". This is a paraphrase of "In this routine it can be assumed that such and such is the case." Its appropriateness stems from the way a programmer codes—setting down successive instructions in terms of what he (the programmer) knows at the time. What the programmer knows at a particular point in a routine is what the routine knows. The following dialogue gives the flavor. (Programmer *A* looking over the shoulder of *B*, who is coding up a routine.) "How come you just added Z5 to the accumulator?" "Because I want . . ." "No, I mean how do you know it's a number?" "All the Z's are numbers, that's the way I set it up." (*B* now puts down another instruction.) "How can you do *that*?" "Because I cleared the cell to zero here at the start of the routine." "But the program can branch back to this point in front of you!" "Oh, you're right; I don't know it's cleared to zero at this point."

The philosophers, of course, have had their own technical development of the concept of knowledge, which did not contribute to the computer science and AI concept, as far as I can tell. Certainly they are distinct concepts. One difference is clearly evident in what is here called *knowledge* is called *belief* by the philosophers, who reserve *knowledge* for something akin to *justified true belief*. Peculiar problems of scholarship are raised when technical communities acquire important new concepts by their practice. For instance, the philosophers have a notion (perhaps even a conceit) called *folk psychology*. It distinguishes the psychology of the folk—of the untutored masses, so to speak—from the psychology as given by science. Is, then, the use of *knowledge* by computer scientists part of *folk philosophy*? It is certainly not what the computer scientists write that counts, but how they use it in their practice. One might equally entertain the notion that the philosopher's use of *knowledge* was *folk computer science*. Except that philosophy got there first, even if by a different route. Now that philosophy has a pseudopod into cognitive science, these two views of knowledge are brought together, mixing in some odd ways.

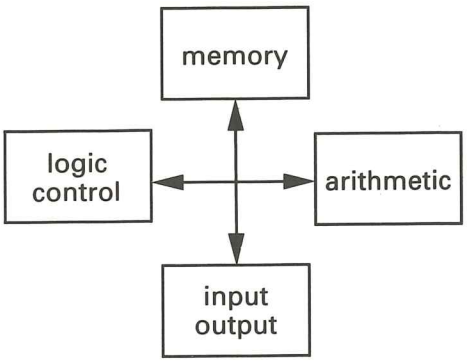

Figure 2-15. Early four-box standard architecture.

MSI to LSI to VLSI.[18] This last, VLSI, has introduced many new notions, such as area constraints and geometrical layout. Nevertheless, the register-transfer level has not changed an iota. There has been no evolution at all in terms of the level structure in computing systems.

We do not understand the force of this invariance. It may in fact be related to the engineering technologies we build or to the conservatism that now inhabits the computer field, with its concern for upward compatibility enforced by economics. Clearly, radically different technologies, such as biological ones, should change the hierarchy somewhat. One of the major challenges in the development of massively parallel connectionist systems is whether they will find a different hierarchical organization. In any event, the computer-systems hierarchy is an important invariant structural characteristic, although its seems to be the only one.

2.7. Intelligence

Intelligence is another central concept of cognitive science. Like the other concepts we have discussed, it has a long history unrelated to the attempts to shape it to the particular needs of cognitive science. Unfortunately, the history of the concept of intelligence is filled with contention, and in a way that the other concepts are not. The contention has spilled in from the larger political and social

18. Medium-scale integration, large-scale integration and very large-scale integration, respectively.

world via the development of intelligence testing and the importance it has assumed in the struggle of minorities and those without privilege. Contention surrounding the other foundational concepts we've covered, such as knowledge and representation, has managed to remain contained, for the most part, to the intellectual [*sic*] disciplines, such as philosophy.

The notion of intelligence is not only contentious, it expresses itself in the existence of a multiplicity of notions (Sternberg, 1985a; Sternberg & Detterman, 1986). These notions all bear a family resemblance, but the differences embody the needs to make intelligence serve different causes, so that synthesis and melding is precisely what is not permitted—which does not prevent recurrent attempts (Sternberg, 1985b). The notion of a single, universal kind of intelligence is set in opposition to multiple intelligences, such as academic intelligence and real-world (or practical) intelligence (Neisser, 1979), which distinction is in the service of arguments over education in the schools and its adequacy for life beyond the school. The notion of a single, universal concept of intelligence is also set in opposition to the relativity of the concept to the culture that defines it, which distinction is in the service of arguments over the dominance of Western ways of thought (Cole & Scribner, 1974). The notion of intelligence as a scientific construct defined entirely by a technology for creating tests is set in opposition to a notion of defining it by cognitive theories and experiment. In this last case, at least, there has been a vigorous and sustained movement to bring the two notions together (Sternberg, 1985a), although not one that has yet fully succeeded.

We cannot thereby simply walk away from the concept. Indeed, there is no reason to do so. Science engages in a continuous process of appropriation and refinement of concepts used elsewhere in society. Some notion is required of a graded ability or power of a mind-like system, which applies in some way to ranges or classes of tasks and ranges or classes of minds to indicate what minds can perform what tasks. There is little doubt about the usefulness of such a concept. It would seem that a theory of mind must contain such a concept. If it doesn't, the theory should tell us why such a notion is not definable or is definable only within certain limits.

In fact, the theory described in this chapter, whose key notions are knowledge, representation, computation, symbols, and architecture, contains within it a natural definition of intelligence:

> A system is *intelligent* to the degree that it approximates a knowledge-level system.

If these other concepts are accepted as we have laid them out, then this definition is what answers to the requirements of a concept of intelligence. It is a theory-bound definition, not an independently motivated concept. Indeed, that is exactly what one wants from a theory of the mind—that it should determine what intelligence is, not that intelligence should be independently motivated.[19]

The formulation of intelligence just given is highly condensed. To understand it, consider three arguments:

1. If a system uses *all* of the knowledge that it has, it must be perfectly intelligent. There is nothing that anything called intelligence can do to produce more effective performance. If all the knowledge that a system has is brought to bear in the service of its goals, the behavior must correspond to what perfect intelligence produces.
2. If a system does not have some knowledge, failure to use it cannot be a failure of intelligence. Intelligence can work only with the knowledge the system has.
3. If a system has some knowledge and fails to use it, then there is certainly a failure of some internal ability. Something within the system did not permit it to make use of the knowledge in the service of one of its own goals, that is, in its own interests. This failure can be identified with a lack of intelligence.

Thus, intelligence is the ability to bring to bear all the knowledge that one has in the service of one's goals. To describe a system at the knowledge level is to presume that it will use the knowledge it has to reach its goal. Pure knowledge-level creatures cannot be graded by intelligence—they do what they know and they can do no more, that is, no better. But real creatures have difficulties bringing all their knowledge to bear, and intelligence describes how well they can do that.

19. I would argue that much of the contention over intelligence has arisen because of its development in psychometrics without connection to the general theory of cognition. This might have been the only way it could have developed during the first part of the twentieth century, given the concurrent hegemony of behaviorism, which served to repress cognitive theories. But the consequence remains.