DRAFT MATERIAL: LIMITED DISTRIBUTION FOR COMMENT.
NOT FOR QUOTATION
File: /afs/cs/project/soar/member/an/wjb/c2fcs.mss
Current version: 31 August 89 11:13
Draft 3 started: 27 Jun 89

Final

The 1987 William James Lectures
UNIFIED THEORIES OF COGNITION

CHAPTER 2
FOUNDATIONS OF COGNITIVE SCIENCE

DRAFT 3.1

Allen Newell

31 August 1989

Departments of Computer Science and Psychology
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Table of Contents

List of Figures

2. Foundations of Cognitive Science

We start with the foundational issues of how to describe the basic structure of cognitive systems. We need some common ground as we go forward, so that we don't become distracted about is knowledge or what is a representation. We will be covering familiar territory. Much attention has been paid to these notions. Some of them have a classical philosophical literature, and all of them have attracted continuing consideration in cognitive science in the last decade, especially as philosophers have finally come to take an interest in the subject. Still, it is worthwhile reworking the familiar. However, it must be recognized there is never much agreement on foundations — they are always less secure intellectually than what they support. Thus, we should view this chapter as primarily building up a common basis for communication for the study at hand, namely, unified theories of cognition.

Despite the disclaimer, I do have something I am attempting to achieve while reviewing these notions. I want to tie them closer to the physical world that is usually the case. Rather than emphasize the abstract character of the various types of systems we will deal with, I will emphasize how they grow out of humans as physical systems in a dynamic world. Theories of human cognition are ultimately theories of physical, biological systems. Our ability to describe human cognition in one way rather than another rests ultimately on their physical and biological nature. Furthermore, their grounding in the world implies additional constraints that shape our theories.

A convenient form for this chapter is to take up, in turn, the major terms that will enter repeatedly into the lectures. We start with the notion of a behaving system, which is just a necessary preliminary. But then we take up knowledge, representation, computational systems, symbols and symbol systems, architecture, intelligence, search and problem spaces, and preparation and deliberation.[12]

2.1. Behaving Systems

I want to take *mind* to be the control system that guides the organism in its complex interactions with the dynamic real world. So, Figure 2-1 shows the environment running through time and the organism running through time, with a series of interactions between the two. Although single transactions can be isolated analytically — where the environment presents and the organism responds, and vice versa — these transactions are embedded in a sequence such that each becomes part of the context within which further actions follow. The mind then is simply the name we give to the control system that has evolved within the organism to carry out the interactions to the benefit of that organism or ultimately to the survival of its species.

Whatever higher point of view might also be valid, mind can be seen to provide *response functions*. That is, the organism takes actions as a function of the environment (in the mathematical sense). If the environment is different, the organism can behave differently, even with the same response function. That is what it means to be in interaction. However, many different response functions occur as the organism goes through time. During the period labeled

[12] In the original lectures, purely for reasons of time, the last two topics — search and problem spaces, and preparation and deliberation — formed the initial segment of Lecture 3. Here, all the foundational material is pulled into one chapter, making it longer and the next one consequently shorter.

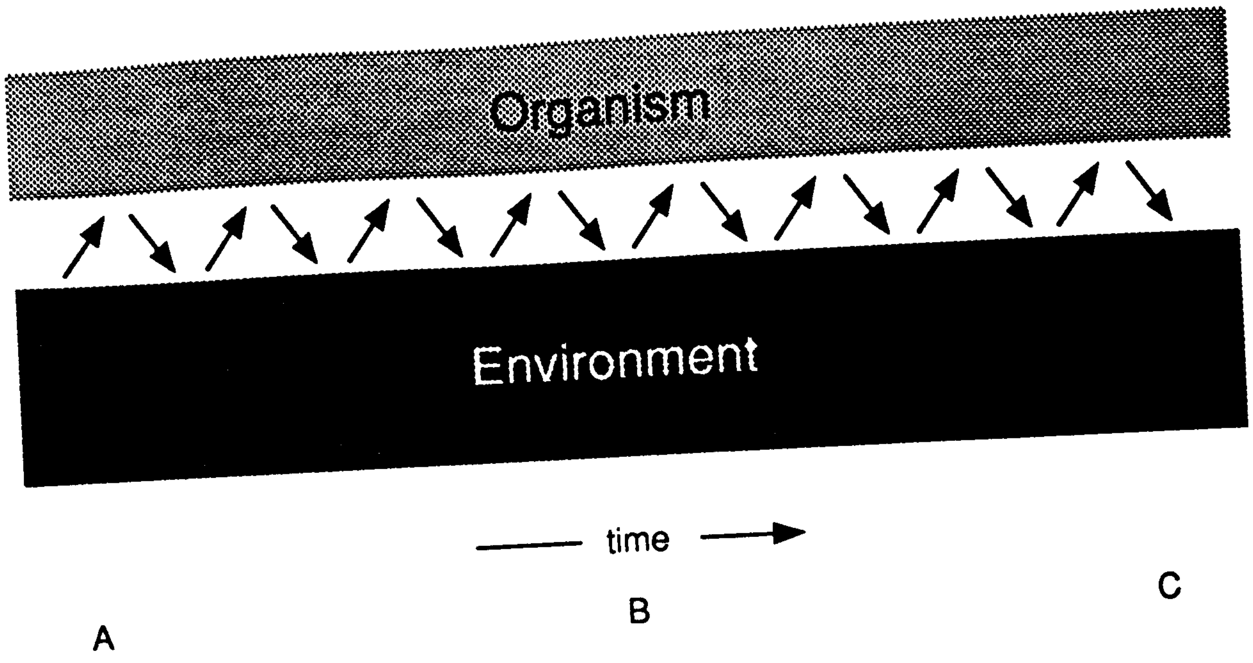


Figure 2-1: Abstract view of mind as a controller of a behaving system.

A in the figure the organism behaves with one response function, during B with another, during C with yet another. There are different response functions in different kinds of situations.

Imagine yourself going through a day. There is one response function when you get yourself out of bed, one when you reach for your clothes, one when you face yourself in the mirror, another when you go to breakfast and talk to your spouse, another when you get in your car and drive to work. Each of those situations is radically different and each calls for a quite different function about how to respond with respect to the environment. One involves beds, floors and covers; another involves mirrors and faucets; another 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 the 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.

The purpose of this section is just to introduce the phrase, *response function*, which will occur over and over again throughout the book.

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 books, taking instructions, doing mathematics, or holding conversations. 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 the 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 sits 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?

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 "

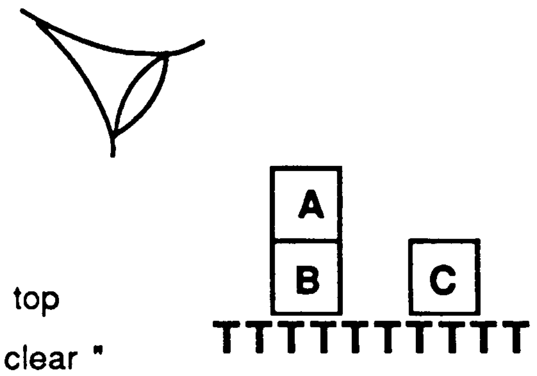


Figure 2-2: The simple blocks world.[13]

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 it will answer the question, rather than (say) simply walking out of the room or doing anything else. The agent's goal (in this case) is something like, if someone asks a simple question, answer them truthfully. We take it that, if X knows something, it can use that knowledge for whatever purposes it chooses. Thus, we calculate: X knows that block A is on top of block B; and X wants to answer the question truthfully; and X has the ability to answer (X can communicate, etc.); 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 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

[13] FigNote: Fig needs fixing.

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. However, 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.[14]

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, though not related to its primary function, e.g., as an item of cost in a budget, as a contributor to floor loading, or as an emblem of being high-tech. All the functional descriptions 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 Kirchoff's laws; logic circuits obey Boolean algebra; the processing in programs obeys its 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 of predictions. But 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 become tiresome to always have to be explicit. Prediction will cover them all.

The descriptions of computer systems form a hierarchy of *levels*, because each higher description is both an abstraction and a specialization of the one below it. Consider electrical circuits. Its medium, current, is the flow of electrons. Its laws, Ohms and Kirchoff's, can be

[14] Our 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, that agent must be described. 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: Ohms law, Kirchoff's law

Electronic Devices

Medium: Electrons
Laws: Electron physics

Figure 2-3: The hierarchy of computer systems.

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 in the simple aggregation that goes from 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 an 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. It is simply another way to describe a system. As a level in the hierarchy, it 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 towards which that content will be used. As a level, there is a medium, namely, knowledge. There is a law of behavior, namely, if the system wants to attain goal G and knows that to do act A will attain 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 media 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 to 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 in order to use it effectively.

Such descriptive schemes, when put forth as 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. The 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 imaging 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, etc. First of all, this someone could certainly be wrong — there could be nothing of the kind in the box. But then, 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 having 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. For an existing system, its behavior can be calculated by knowing 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, where the 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.

moved prior draft ↓

Figure 2-4: The knowledge-level system.[15]

Characterizing knowledge as the medium of a system level, which is just one system level among many, constitutes a particular 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 system terms at all, but simply proceeds to inquire after its validity and certainty. However, Daniel Dennett's (1978, 1988) notion of an *intentional system* is substantially the same as a knowledge-level system. Actually, the key concept for Dennent is that of the *intentional stance*, which is the way the observer chooses to view or conceptualize the agent.[16]

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.

That this theory of knowledge arose without specific authorship — without a specific inventor

It is use to reflect on

[15] FigNote: To be revised.

[16] It 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, Newell, 1988b).

or discoverer — is worth comment. 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. 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 get s the credit. However, the case in hand (and others such as symbols and architecture ~~coming later in the lecture~~) doesn't fit this frame.

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 such a 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 seen as 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).[17] 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 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 p403):

> "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 its 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*, ~~but which~~ is called *belief* by the philosophers, who reserve *knowledge* for something

[17] 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).

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 differently. Now that philosophy has a pseudopod into cognitive science, these two views of knowledge are brought together, mixing in some odd ways.[18]

2.3. Representation

Knowledge abstracts from representation. However, knowledge must be represented in some fashion in order to be used, told, thought, etc. This may seem like a special philosophical statement, justified by some particular argument. But it is not. It is simply the proposition that knowledge-level systems are simultaneously systems describable at lower levels and that systems that exist in the physical world have physical descriptions. Thus, to use Figure 2-3 as a guide, a knowledge-level system that exists in the physical world also can be described as a programmed system, a register-transfer system, a logic system, an electrical circuit, and an electronic device. Of course, the hierarchy of that figure is not necessarily the only hierarchy that can have knowledge systems at the top. The figure exhibits one hierarchy that we know of that does include knowledge systems. Other hierarchies might well exist. Indeed, in the case of humans, who are describable as knowledge-level systems better than any of the computer systems we have around, the hierarchy must be quite different as it makes its way down to the biological substrate. Establishing the existence of alternative hierarchies that support a knowledge level is a scientific task or an engineering task, depending on whether we discover the hierarchy by the analysis of existing systems or invent it by the design and construction of hitherto nonexisting systems.

Let us turn, then, to the nature of representation, which is another fundamental concept in cognitive science. Let us start by being specific. A standard way of representing knowledge is with a logic. Figure 2-5 provides such a representation for the blocks world. The representation consists of the expressions in the figure. There is a predicate *block* that can be used to assert that A is a block, B is a block, C is a block and the table T is not a block. Another predicate *on*, can be used to assert that B is on the table T, A is on B, and C is on T. Another predicate *clear* is defined in terms of *block* and *on*: x is clear if and only if x is a block and, if y is any block, then it is not on x. From this information, it can be inferred that block B is not clear. Thus if expressions 1-3 in Figure 2-5 represent what X knows, then X also knows that B is not clear. And if we couple this with knowledge of X's goal, Y's request, and X's ability to communicate, then we can predict that X will tell Y that B is not clear.

One might be tempted to object that X knows not(clear B) directly from observation, and not

[18] Another way to express this same confusion is to ask whether the informants of anthropologists should be co-authors of the anthropological studies for which they inform. Anthropologists have generally answered in the negative, and taken the credit of creation of scientific knowledge to themselves alone (Marcus & Fischer, 1985).

1. (block A), (block B), (block C), not (block T),

2. (on B T), (on A B), (on C T)

3. (clear x) iff (block x) and (y) (block y) implies not (on y x)

Infer

4. not (clear B)

Figure 2-5: The blocks world in logic.[19]

indirectly via some inference. But this .confuses the knowledge that X has ~~from~~ with the representation that X has of that knowledge. Certainly, X has some representation of this knowledge. But we do not know that it is the logic representation of Figure 2-5. In fact, if X is a person and you are a psychologist, you'll bet that it isn't. All we assert is that this logic representation tells us the knowledge that X has. Nothing is said about the form of that representation, certainly not that the individual expressions in Figure 2-5 correspond to anything particular in X's representation.

It is important to be clear that a logic is just a way of representing knowledge. It is not the knowledge itself. To see this, let K(1) be the knowledge represented by expression 1 in the figure. Thus, by writing down expressions 1, 2 and 3, as the knowledge that X has, we certainly mean that X has knowledge K(1) and K(2) and K(3). One might think that it makes no difference whether we talk about the expressions or the knowledge — they are in one-to-one correspondence. But that is not the only knowledge X has. X also has K(4). K(4) certainly does not correspond to any expression that we originally wrote down (namely, 1, 2 and 3). X had K(4) because expression 4 can be inferred from expressions 1, 2 and 3 (and the rules of the logic).

The general situation for logic can be stated easily. If X has the knowledge represented by a conjunction of expressions K(1, 2, ..., M), and expression N follows from 1, 2, ... M, then X has the K(N), the knowledge of expression N. Since, in general there are an unlimited number of expressions that follow from a conjunction of expressions, X's knowledge is correspondingly unlimited. It certainly cannot be written down in its entirety. So there is no way in which one can identify ~~the~~ knowledge of X with the representation of that knowledge. Instead, what a logic lets us do is ~~write~~ represent the knowledge of X as a finite set of expressions plus an process (the inference rules of logic) for generating the infinite set of other expressions which comprise X's total knowledge.

It is also important to see that logic is just one of many different ways of representing knowledge. It was our *choice* to use logic to represent X's knowledge. It might or might not be a good way of representing what X knows. It might be that X would show that it knew ~~the~~ K(1) and K(2) and K(3), but would not exhibit any knowledge of K(4). Then this logic representation would not describe X. It would be too powerful, implying that X knew things it didn't. We might try to fashion a new representation for X's knowledge by positing that X knew only what

[19] FigNote: Add "Posit" as a heading to lines 1-3.

The ~~such~~ a revised scheme

log The

is expressed in ~~the~~ exact set of expressions that are written down, where no additional inferences are permitted. For some purposes this might be useful, though with logics it is isn't, since a logic carries with it a set of rules of inference and the notion of the free use of the rules.[20] If we accept our earlier story, X did know K(4), so ~~in this case it~~ would be too weak, implying that X didn't know things it did. We might then cast about for other ways of representing knowledge that would provide an accurate knowledge-level description~~s~~ of X. ~~It must be some~~ no guarantee, of course, that we can ever find such a

There is no guarantee X might not be represented as a K-l syst

Although logic is only one of many ways of representing, it has many nice properties that reflect general properties of bodies of knowledge. Two bodies of knowledge can be combined to form a single body of knowledge, as in the acquisition of some knowledge into an existing system. This corresponds to the conjunction of the two sets of expressions in the logic. If one adds to a body of knowledge exactly the same knowledge, then no new knowledge is obtained. STET This is reflected in conjunction being idempotent: (A and A) ~~if and only if~~ A. A particularly important feature is the existence of theorems of completeness. In logics, such as the first-order predicate calculus, essentially any knowledge can be represented (LogicSufficiencyXX-2). Such sufficiency is bought at the price of how that knowledge is represented. In fact, there is much work in artificial intelligence and some in philosophy in how to represent things in logical calculi. Much of the difficulty turns out to be that we don't understand what we want to represent, not that we can't find first-order ways of doing it. Another part of the difficulty is that the representation is often very awkward and indirect. Thus much research goes into finding alternative forms of logics (model logics, higher-order logics, sorted logics, etc.) which are easier to use, more perspicuous, etc.

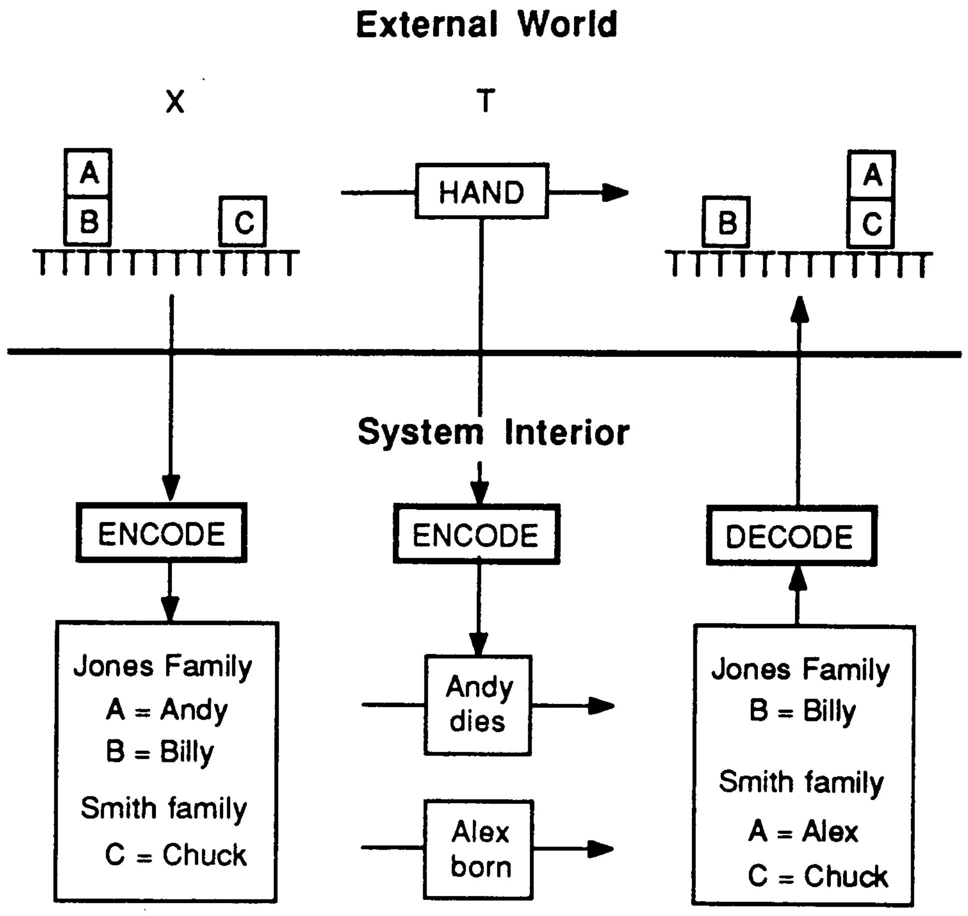
set theory

We can become clearer about what is involved in representation, by pursuing how else we might represent our simple blocks world. The general situation is shown in Figure 2-6. There is an external world, in which the blocks world occurs. This time we have indicated an action (the transformation T) that moves the block A from atop block B to atop block C.[21] Besides the external world, there is the system, in whose interior is another situation, which is to represent the blocks world. This interior situation is equally part of the total world, of course, and consists of some physical system that goes through transformations as well, to produce another interior situation. The issue is what is the nature of the interior system for it to be a representation of the particular blocks situation in the external world.

I've chosen a rather unfamiliar way of representing the blocks world in Figure 2-6, namely in terms of families. Each stack of blocks corresponds to a family — the Jones family and the Smith family — with the children in a family being the blocks in order of age, youngest on top. The transformation is that people die and get born. If Andy Jones dies and Alex Smith is born, then the Jones family only has Billy and the Smith family has Alex and Chuck. To ask a question, such as whether a given block, say B, is clear, is to ask who is the youngest child in a family. In the initial situation, Billy is not the youngest, although after Andy dies, he becomes the youngest.

[20] Recently there has been some interest in using logics with restricted rules of inference, so that bounds can be given on the time to make inferences (Levesque, 1986).

[21] We could have remained faithful to the prior example of whether block B was clear, but it would have been somewhat confusing, because it involves only a change in the knowing agent, not in the world.



A block is clear if it corresponds to the youngest child

Figure 2-6: A representation of the blocks world.[22]

There is a clear correspondence between the external blocks world and family situations. Some of the properties correspond (being clear, and being the youngest), but not all (are the blocks square, or are the children boys or girls?). Some of the transformations correspond (moving blocks, and borning/dying), but not all (blocks falling over, and the children growing older).

The representation is somewhat odd. I chose it to avoid the obvious ways we would create such representations, if we were doing it on paper or in a computer. However, it's actually not as far fetched as you might think. According to a long tradition in anthropology (Laboratory of Comparative Human Cognition, 1982), when one attempts to get members of primitive tribes to reason on isolated, abstract logical tasks, they tend not to work with the abstract situations, but to deal only with concrete situations with which they are thoroughly familiar. In fact, that is true with people in our own society. People are better on syllogisms if they are cast in concrete terms rather than expressed abstractly. For instance in reporting an experiment on syllogistic reasoning (Johnson-Laird, 1983 p. xxx), it was noted that one person had to be discarded because she said she couldn't imagine any particular concrete situation that corresponded to a particular syllogism, so was unable even to work on the task (in fact, the situation sought involved a family).

Baron

Let's describe ~~[illegible]~~ abstractly what is going on in Figure 2-6. The original external situation

[22] FigNote: In all these figures, replace TTT with rectangle.

is *encoded* into an internal situation. The external transformation is also *encoded* into an internal transformation. Then the internal transformation is *applied* to the internal situation to obtain a new internal situation. Finally, the new internal situation is *decoded* to an external situation. Suppose that the external situation is the same as the situation produced by the external transformation. Then the internal system — the encoding process, the internal situation, the internal transformation, and the decoding process — has successfully been used as a representation of the external situation.

This is the essence of representation — to be able to go from something to something else by a different path when the originals are not available. We can cast this as a general law:

> *The representation law:*
> Decode(encode(T)(encode(X))) = T(X)
>
> ae-
> ~~W~~here X is the original external situation and T is the external transformation.

This is called *the* representation law because it is the general form. Actually, there are a myriad representation laws, one each for the myriad particular encode-apply-decode paths that represent an external path. Each encode-apply-decode path is unique exactly as required to reflect the aspects of the external path that are represented. The processes of encoding and decoding are part of the path, so they become an essential part of the law. There can be multiple forms of encoding, for there are many different situations which can lead to the same encoded situation. The same external situation can (on occasion) be viewed, heard, read about, or felt. Likewise, there can be multiple forms of decoding, for there are many different ways to interact with an external situation. The same external situation can be identified as another (i.e., tested), selected out, constructed, or described (leaving the actual contact to someone else). The central claim behind the representational law is that if anything is used as a representation, then a law of the above form will be the essential ingredient of the support for why the internal situation represents.

For some arrangement to be a representational system carries with it one other requirement. Namely, the internal system has to be controlled in some way. The application of encoding, internal transformings and decoding must be executable at will, or at least sufficiently at will to serve the purposes of the organism. It is not necessary that freedom be complete, but representations must be at the service of other activities of the organism, and hence must be evocable at appropriate times and places. If the family representation in Figure 2-6 were to depend on some real family, say recalled or being perceived concurrently (or worse, if it required waiting until children were actually born and died), it would not pass this requirement. On the other hand ~~However~~, the representational system need not be in the interior of the organism, although Figure 2-6 makes it appear that way. A representation can just as well occur in some exterior field, so long as it can be controlled — a scratch pad, an abacus, a computer, or even a friend.

This view of representation leads naturally to the following approach to representational systems. Given a task for the organism that requires a representation — that is, that requires that the organism produce a response function that depends on aspects of the environment that are not immediately available. Then, the problem for the organism (or for evolution, it makes no difference) is to find the right kind of material with the right properties for encoding and decoding and the right dynamics for the transformation. If these can be found and brought

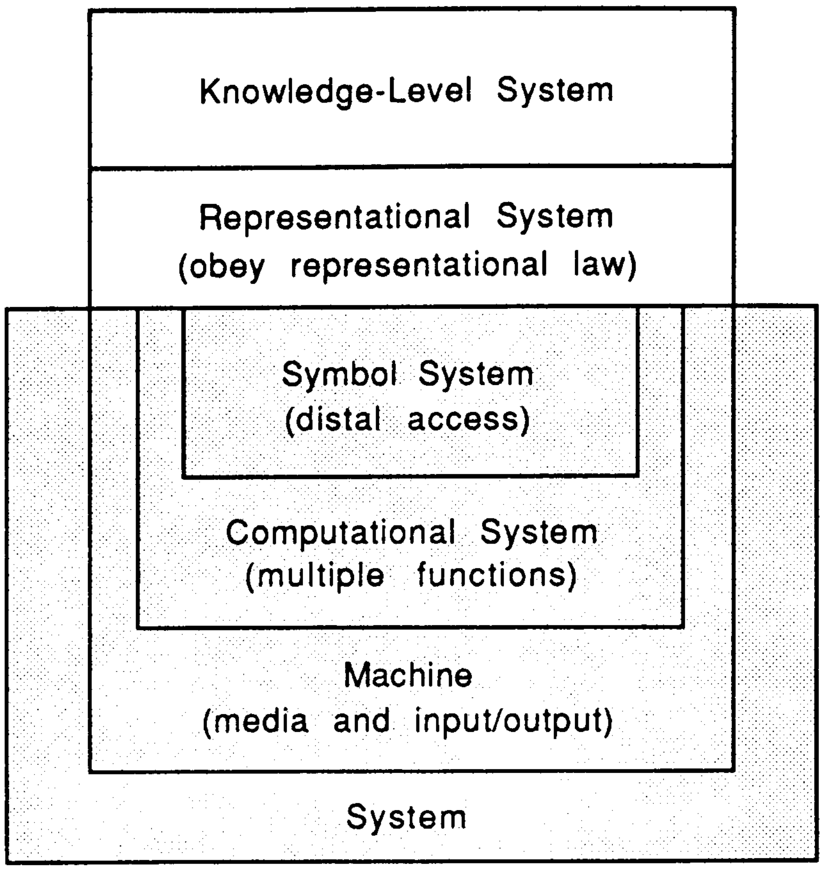


Figure 2-22: The different types of systems.[47]

cannot be described described at the knowledge level.

Representational systems must be realized in some structure, so the two rounded areas intersect the four types of system structures. Representational systems with substantial scopes for domains and goals require symbol systems for their realization, but as scope diminishes then structures of lesser capability can be used as well. The areas of intersection with these more general system structures become smaller and smaller, indicating their more specialized character. There are of course many structures of each type that do not satisfy representational laws, hence are not about the external world, hence to do support active adaptation.

[47] FigNote: Totally new figure.