

Marr's Three Levels: A Re-evaluation

Ron McClamrock
University at Albany, SUNY
[Published in *Minds and Machines*, May 1991]

In recent work in the theoretical foundations of cognitive science, it has become commonplace to separate three distinct levels of analysis of information-processing systems. David Marr (1982) has dubbed the three levels the *computational*, the *algorithmic*, and the *implementational*; Zenon Pylyshyn (1984) calls them the *semantic*, the *syntactic*, and the *physical*; and textbooks in cognitive psychology sometimes call them the levels of *content*, *form*, and *medium* (e.g. Glass, Holyoak, and Santa 1979).

But a mistaken distinction (however ubiquitous) by any another name will do no more work. I think that there is something promising but nonetheless misleading about the "three-levels" analysis of cognitive systems. In what follows, I'll try to show this by in part by bringing to bear some of what the recent philosophy of science has to offer on the analysis of complex systems. First I'll look briefly at a few general ideas in the philosophy of science on this general topic; I'll then turn to an explicit look at the Marr-style three-level view and its shortcomings, and suggest how this account might be revised to avoid these problems.

1. Levels of organization

It has become a central tenet of the current conventional wisdom in the philosophy of science that complex systems are to be seen as typically having multiple levels of organization. The standard model of the multiple levels of a complex system is a rough hierarchy, with the components at each ascending level being some kind of composite made up of the entities present at the next level down. We thus often have explanations of a system's behavior at higher (coarser-grained) and lower (finer-grained) levels. The behavior of a complex system -- a particular organism, say -- might then be explained at various levels of organization, including (but not restricted to) ones which are biochemical, cellular, and psychological. And similarly, a given computer can be analyzed and its behavior explained by characterizing it in terms of the structure of its component logic gates, the machine language program it's running, the Lisp or Pascal program it's running, the accounting task it's performing, and so on.

Higher-level explanations allow us to explain as a natural class things with different underlying physical structures -- that is, types which are *multiply realizable*. [See, e.g., Fodor (1974), Pylyshyn (1984), esp. chapter 1, and Kitcher (1984), esp. pp.343-6, for discussions of this central concept.] Thus, we can explain generically how transistors, resistors, capacitors, and power sources interact to form a kind of amplifier independent of considerations about the various kinds of materials composing these parts, or account for the relatively independent assortment of genes at meiosis without concerning ourselves with the exact

underlying chemical mechanisms. Similar points can be made for indefinitely many cases: how an adding machine works, an internal combustion engine, a four-chambered heart, and so on.

This strength of capturing generalizations has many aspects. One is of course that higher-level explanations typically allow for reasonable explanations and predictions on the basis of far different and often far less detailed information about the system. So, for example, we can predict the distribution of inherited traits of organisms via classical genetics without knowing anything about DNA, or predict the answer a given computer will give to an arithmetic problem while remaining ignorant of the electrical properties of semiconductors. What's critical here is not so much the fact of whether a given higher-level phenomenon is *actually* implemented in the world in different physical ways. Rather, it's the *indifference* to the particularities of lower-level realization that's critical. To say that the higher level determination of process is *indifferent* to implementation is roughly to say that if the higher-level processes occurred, regardless of implementation, this would account for the behaviors under consideration.

This general view of complex systems depends on idealizing about the behavior of particular lower-level structures; viewing them simply in terms of their normal input/output functions and their local contribution to the behavior of the larger system rather than in terms of the details of their internal structures. Generality of explanation is achieved by taxonomizing subsystems via input/output functions and allowing and indifferently with respect to the internal structure by which they might produce that function. The computer is again the easiest illustration: The analysis of the behavior of a given computer in terms of, say, the LISP program it's running is a perfectly good one; but it leaves totally unspecified how a given primitive Lisp function (such as *car(students)* -- i.e. "give me the first item on the list *students*") is calculated. The Lisp program is completely compatible with any way you like of representing the lists in memory, or even with different underlying machine architectures (e.g. it could be implemented on a Von Neumann machine, a Turing machine, or whatever).

The question of exactly when such idealization about the behavior of components is appropriate is a difficult one. But clearly at least this constraint is critical: The idealization must be close enough to the real behavior of the system to adequately capture it in some normal range of working conditions. Thus, an adder which generates overflow errors only above $10^{1000000}$ will typically be perfectly well idealized simply as an adder, but one which does so anywhere above 7 probably will not. Exactly how close the real behavior and the ideal must match may be a question with no perfectly general and systematic answer.

2. Function and context

The idea of specifying the components of a larger system in terms of their overall functional role with respect to that embedding system is a central aspect of our general framework of explanation which plays several key roles in the understanding of complex systems. The functional properties of parts of complex systems are *context-dependent* properties -- ones which depend on occurring in the right context, and not just on the local and intrinsic properties of the particular event or object itself. The phenomenon of context-dependence of course shows up often in the taxonomies of various sciences. Examples abound: The position of a given DNA sequence with respect to the rest of the genetic material is critical to its status as a *gene*; type-identical DNA sequences at different loci can play different hereditary roles -- be different genes, if you like. So for a particular DNA sequence to be, say, a brown-eye gene, it must be in an appropriate position on a particular chromosome. Similarly for a given action of a computer's CPU, such as storing the contents of internal register A at the memory location whose address is contained in register X: Two instances of that very same action might, given different positions in a program, differ completely in terms of their functional properties at the higher level: At one place in a program, it might be "set the carry digit from the last addition", and at another, "add the new letter onto the current line of text". And for mechanical systems -- e.g. a carburetor: The functional properties of being a choke or being a throttle are context-dependent. The very same physically characterized air flow valve can be a choke in one context (i.e. when it occurs above the fuel jets) and a throttle in another

(when it occurs below the jets); whether a given valve is a choke or a throttle depends on its surrounding context. By "contextualizing" objects in this way we shift from a categorization of them in terms of local and intrinsic properties to their context-dependent functional ones.

One critical role that such functional analyses play is that of illuminating the lower levels of the system: We quite typically need to know *what* a complex system is doing at a higher level in order to find out *how* at the lower level it accomplishes that task -- that is, we often need to know the function of complex system being analyzed to know what aspects of structure to look at. Not only is there typically a mass of detail at the lower levels which must be sorted through, but salience at the lower levels can be misleading, and can fail to pick out which lower-level properties are important to understanding the overall working of the complex system. So, to take a standard example: If you think that the heart is basically a noisemaker, the lower-level properties which will seem most significant might be things like the resonant frequency of the various chambers, or the transient noises created by the movement of the various valves. Or if you think of a computer as a radio signal emitter, you will see as most salient the high-frequency switching transients of the transistors and the exact frequency of clock signals, and basically ignore the difference between 0's and 1's represented by different DC voltages. Understanding the behavior of a complex system requires knowing which aspects of the complex mass of lower-level properties are significant in making a contribution to the overall behavior of the system; and this depends on having some sense of the higher-level functioning of the system. (Oatley 1980 gives a nice illustration of some of these points by discussing a thought-experiment where we imagine finding a typical microcomputer on trip to a distant planet and -- not knowing what it is -- apply various sorts of research techniques to it. The example does a nice job of illustrating some of the biases built into various methods and in particular bring out the importance of putting lower-level data into a higher-level framework.)

A couple of observations about function and context-dependence which will be important to us later: One is that considerations about context-dependence can and should arise at more than one level of analysis of a complex system, and may have quite different answers at the different levels. For example, the properties of DNA sequences as objects of *chemistry* depends only on their local physical structure. But their properties as *genes* depend on their overall contribution to the phenotype; and what contribution they make to the phenotype is highly dependent on context -- on where the sequence is in relation to the rest of the genetic materials, and on the precise nature of the coding mechanisms which act on the sequences. Or from the point of view of a LISP program, a function like (car(list)) (i.e. 'get the first item on the list named "list"') is a local characterization of that action, whereas an appropriate functional characterization of that operation in a particular case might be "get the name of the student with the highest score on the midterm". But from the machine language point of view, the Lisp characterization would be a functional analysis of some sequence of machine language instruction -- instructions which might play a different role in some other context. (Failing to notice this level-relativity of issue of context dependence has had widespread consequences in the philosophy of the higher-level sciences. For a clear example of this, see McClamrock (in press), which shows how Fodor's defense of "methodological individualism" (Fodor 1987) relies on exactly this error.)

The other observation is that contextualizing or de-contextualizing can be done without a concurrent shift in level -- that is, we might reinterpret the functions of parts against a broader background of the system without at the same time shifting the level size of the parts. The shift between assembly-language and machine-language is roughly like this. The degree of abstraction is essentially the same: assembly-language is essentially a one-to-one translation of the raw numbers which the CPU operates on into mnemonics for the functional role the number is playing in that instance. Thus, at one point in the program, a given raw number *n* might be the op code for a command to the CPU, and at another be data to be operated on (e.g. added, moved, etc.). The size or grain of the functional units has not changed, but the functional/contextual characterization has.

3. The three levels

In chapter 1.2 of *Vision*, David Marr presents his variant on the "three levels" story. His summary of "the three levels at which any machine carrying out an information-processing task must be understood":

- *Computational theory*: What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?
- *Representation and algorithm*: How can this computational theory be implemented? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?
- *Hardware implementation*: How can the representation and algorithm be realized physically? [Marr (1982), p. 25]

As an illustration, Marr applies this distinction to the levels of theorizing about a well-understood device: a cash register. At the *computational* level, "the level of *what* the device does and *why*", Marr tells us that "what it does is arithmetic, so our first task is to master the theory of addition" [ibid, p. 22]. But at the level of *representation and algorithm*, which specifies the forms of the representations and the algorithms defined over them, "we might choose Arabic numerals for the representations, and for the algorithm we could follow the usual rules about adding the least significant digits first and 'carrying' if the sum exceeds 9" [ibid, p. 23]. And at the implementational level, we face the question of how those symbols and processes are actually physically implemented; e.g., are the digits implemented as positions on a ten-notch metal wheel, or as binary coded decimal numbers implemented in the electrical states of digital logic circuitry?

The relationships between Marr's levels preserve many of the properties of the conventional inter-level relationships in complex systems. For example, there is the multiple realizability of a higher-level structure by various lower-level structures. For a given computational theory, "there is a wide choice of representation.... [and] even for a fixed representation, there are often several possible algorithm for carrying out the same process" [ibid, p. 23]. And "the child who methodically adds two numbers from right to left, carrying a digit when necessary, may be using the same algorithm that is implemented by the wires and transistors of the cash register... but the physical realization of the algorithm is quite different" [ibid, p. 24].

Zenon Pylyshyn's variant of the "three levels" view [Pylyshyn (1984)] is similar in many respects. As he says, the "main thesis" or "basic working hypothesis" of his book *Computation and Cognition* is that within the study of cognition, "...the principle generalizations covering behavior occur at three autonomous levels of description, each conforming to different principles. These principles are [here] referred to as the biological (or physical) level, the symbolic (or syntactic or sometimes the functional) level, and the semantic (or intentional) level" [ibid, p.259]. Or again: "...we will see that there are actually two distinct levels above the physical or neurophysiological) level -- a representational or semantical level and a symbol-processing level" [ibid, p.24]. Thus we have as levels the *biological* (Marr's *implementational*), the *symbolic* or *syntactic* (Marr's *algorithmic*), and the *semantic* (Marr's *computational*). And as for Marr, the distinction of levels is seen as having some of its standard roles -- e.g. being used "to account for certain kinds of generalization and to give a principled account of certain constraints and capacities" [ibid, p. 39].

One of the central means given by Pylyshyn for characterizing the semantic level account is of particular interest here. As he says, semantic level explanations "make substantive reference to entities or properties that are not an intrinsic part of their state description, for example, *numbers* or *need for assistance*" [ibid, p. 25]. Of course, he has in mind the fact that this intrinsic dependence of characterization on things external to the state itself is a standard mark of the *intentional* -- being defined in terms of its *object* rather than its intrinsic form. Again, we see *context-dependence* or *functional* characterization at the center of the relationship between the algorithmic/syntactic account and the computational/semantic account.

4. Shifts in grain, shifts in context

Unfortunately, the accounts share something else --- a central confusion. Both Marr's computational-to-algorithmic transition and Pylyshyn's semantic-to-symbolic one run together two distinct sorts of changes: One kind of change is a further decomposition and lessening of the degree of abstraction of the activities, where components in the higher-level explanation are further decomposed. This is, I think, the central idea in the general notion of a shift in level of organization. The other kind of change is one from *functional* properties to *local* ones by de-contextualizing (e.g. from talk of numbers and addition to the symbols representing those numbers and processes defined over them as syntactic objects).

Now if we were to take the "three levels" view as making a claim about the actual number of levels of organization in cognitive systems, it would be a very substantive (and I think false) empirical claim. It would be claiming that cognitive systems will not have any kind of multiple nesting of levels of organization. Why should our explanatory framework for cognition have this kind of limiting assumption built into it? Of course, it's imaginable that the brain is special in this way; someone might possibly try to offer some reason to think that evolved systems -- as opposed to designed ones -- don't have multiply nested levels of processing. (Simon (1981) of course argues exactly the opposite point -- that we should *expect* hierarchies from evolution.) But the number of autonomous levels of organization of a system would seem to be an empirical fact about each particular type of system we consider. We have in hand no persuasive reason for thinking that the brain is somehow special and different in this respect.

But neither would it be appropriate to take the relationship between Marr-type levels as a distinction in contextualization. To do so would be to ignore the fact that the relationship between the abstract task structure and the particular procedure performing it (of the many such procedures possible) surely exhibits a distinction in degree of coarseness of grain (or true "level") -- whatever else may or may not go into that relationship.

And finally, to take it as the conjunction of the two would be to ignore the difference between these two importantly different kinds of explanatory shift. One, a change in the degree of functional decomposition (i.e. of the true level of organization), which might be done any number of times for some particular information-processing system; and the other, a change of functional characterization, where at the "higher level" the processes and symbols are characterized less in terms of their intrinsic state and more in terms of the overall functional role they play in the working of the device. And clearly these two shifts needn't come together, as the example of the shift between machine-language and assembly-language given earlier illustrates.

These same general points should be seen as applying to the distinction between the algorithmic and the implementational "levels" as well. The transition between these is a matter of (1) further functional decomposition which specifies the internal (physical) structure of the operations taken as primitive in the specification of the algorithms, and (2) leaving behind the characterization of the activity in terms of symbolic processes and replacing it with a characterization of the underlying physically specified mechanisms. Of course, the implementation of the primitives of any given algorithmic level may well be itself non-physical -- implementation in another virtual machine rather than directly in the hardware.

5. Perspectives on levels

In reconstructing the 'three levels' account, it's important to note that it is the characterization of account of representation and process (the *algorithmic/representational* or *syntactic* level, in the current lingo) which is typically seen as the central goal of cognitive or information-processing explanation.

Often, the interest in the other two levels is primarily in how they might bear on and elucidate the algorithmic/representational level. Even Marr, who focused centrally on the computational level and assigned it a kind of priority in research, often motivated and justified the pursuit of computational level theory largely placing it on the path to the representation and algorithm; as he put it, "an algorithm is likely to be understood more readily by understanding the nature of the problem being solved than by examining the mechanism (and the hardware) in which it is embodied" [Marr (1982), p. 27].

Taking seriously the explanatory centrality of explanation in terms of representation and algorithm, we might try to recast what's importantly correct in Marr's account in something like the following way: The number of actual algorithmic levels of organization in any given information-processing system (including the brain) is an entirely empirical matter about that particular system. It could of course be *zero* (as some *eliminativists* would have us think), but it could also be far more than one (as in the nested virtual machines in a real computer). But for each of those levels of organization or decomposition, there are three *perspectives* that we can take toward it -- or if you prefer, three general kinds of questions that we can pose about it, or three kinds of explanations of it we might try to give: questions about that structure itself; questions about the functional, context-dependent properties of the parts and relations in that structure and their contribution to the functioning of the system as a whole; and questions about the implementation of the primitive parts of that algorithmic structure. Or to put it in a way even closer to Marr's, we might see the three perspectives of *algorithm*, *content of computation*, and *implementation* as having something like the following questions associated with them:

- *Format and algorithm*: What is the syntactic structure of the representations at this level, and what algorithms are used to transform them? What is the real structure of the virtual machine? What's the program? From this perspective, the questions are explicitly information-processing questions. Further, it's this level of functional decomposition of the system which specifies the level of organization with which we are currently concerned, and to which the other two perspectives are related.
- *Content, function, and interpretation*: What are the relational or global functional roles of the main processes described at this level? What tasks are being performed by these processes, and why? These are centrally questions about the interpretation and global function of the parts and procedures specified in our algorithmic analysis.
- *Implementation*: How are the primitives of the current level implemented? By another computationally characterized virtual machine? Directly in the hardware? How much decomposition (in terms of kinds of primitives, structures, abilities, etc.) is there between the current level and what is implementing it? How much of the work is done by the postulated primitives of this level as opposed to being done explicitly by the analyzed processes? The shift from algorithm to implementation is thus centrally one of levels of organization or functional decomposition; i.e. of what happens when we try to move down a level of organization.

Given the focus encouraged by Marr and Pylysyn on counting here, one might however ask at this point why there are in this case *three* perspectives instead of *five*. After all, with two at least somewhat independent ways to make transitions away from the central explanation (by level of organization and by contextualization) and two directions to move ("up" and "down", intuitively), there might to be *four* explanatory shifts available, and so *five* perspectives.

But the fact that the central level is taken to be *syntactic* in nature actually constrains the possibilities here. The processes at that level are defined in virtue of local formal properties. Thus, in moving "down" in explanatory grain, de-contextualization can't occur before decomposition. If the algorithmic account has been specified in terms of the local, syntactic properties of the representations and the algorithms that operate on them, then the global or contextual properties of these structures will already have been eliminated from that account. And the general priority of top-down information flow in research discourages the other

transition -- up by grain without contextualization. Without a contextualization of the parts at the algorithmic "level", we won't be able to see which parts should be lumped together to provide higher-level functions. That is to say: Without an account of higher-level *function*, we won't know which aspects of which lower-level structures should show up and which should be screened off from showing up at the next higher level of organization of the system.

Thus, although there may be no set of three special levels of explanation of cognition, the original distinction might be recast to provide three natural classes of questions and issues which arise at any information-processing level of explanation. Doing this may well capture what seems correct about the analysis, while both (a) leaving behind the unwarranted constraints it placed in its more standard version (i.e. only three levels, and implementation as necessarily physical rather than symbolic), and (b) separating out the two kinds of shifts (i.e. of level and of interpretation) which had been run together.

6. Concluding remarks

As with any fairly abstract theoretical distinction, the eventual proof of the pudding will be in the eating. Thus, how this re-characterization of the "three levels" distinction should be viewed will depend critically on the overall role it plays in the explanations offered and debates about them. Where then might this conclusion come into play?

One relevant context might be that of the current debates about connectionism and its potential conflict with the more classical account of representation. First, emphasizing the possibility that actual *levels* of even algorithmic/representational structure may well be multiply nested encourages the possibility of a kind of compatibilism, where one use of connectionist models is as an implementational account of the primitive processes of some higher --- and potentially more "classically representational" --- algorithmic structure. Rather than reducing classical models to "epiphenomena" or connectionist models to "mere implementation" [Smolensky 1988], there might well be plenty of algorithmic explanation to go around. In fact, at least one such debate in this area (between McClelland and Rumelhart (1985) for the connectionists and Broadbent (1985) for the traditional cognitivists) focused explicitly on how Marr's tri-level view would classify the projects in they were involved. Much of the debate reduced to Broadbent's insistence that the more standard explanations in cognitive psychology are the "algorithmic" (thus assigning the lower-level connectionist model the role of "mere" implementation) being countered by McClelland and Rumelhart's claim that it was their own work which was at the algorithmic level (thus leaving Broadbent's model in a direct -- losing -- competition with theirs). But surely there's no point in fighting over who's *really* the algorithmic level. The relationship between connectionism and more standard cognitive models is a complex and interesting one, but seeing it as any kind of simple competition for the one "true" level of representation and algorithm -- a view which the standard trilevel account has in this instance encouraged -- is surely to be avoided.

This is not a vote for vapid eclecticism, however. There are of course real challenges that explanatory account of the behavior of complex systems must meet regardless of the level of organization they attempt to capture. An explanation at any level must face the challenge from "below" that it may have failed to provide an idealization about the coarser-grained structure of a complex system which approximates the actual behavior of the system to a significant enough degree. And from "above", it must face the question of whether it has left behind critical generalizations [see Pylyshyn (1984), chapter 1], or buried the important processes in a mass of irrelevant implementational detail [see Kitcher (1984), esp. pp. 347-9].

The domain of psychological explanation is one where the imposition of "philosophical" constraints on theory may be especially critical because of the nebulous nature of the subject matter. Unfortunately, it has also been a domain where those ideological constraints have often misled more than they have helped -- witness behaviorism, or introspectionism. Given this, we should be particularly vigilant about the possibility of unreasonable and perhaps even

unnoticed constraints slipping in and adding to the confusion -- there is, after all, plenty of that to go around already.

Thanks to William Wimsatt, Mike Swain, Lance Rips, Stuart Glennan, Dan Gilman, Rob Chametzky, and Marshall Abrams for their invaluable help through our conversations and their comments on earlier drafts.

References

- Broadbent, D. (1985), 'A Question of Levels: Comment on McClelland and Rumelhart', *Journal of Experimental Psychology: General* 114, pp.189-192.
 - Fodor, J.A. (1974) 'Special Sciences, or the Disunity of Science as a Working Hypothesis', *Synthese* 28, pp. 97-115.
 - Fodor, J.A. (1987), *Psychosemantics: The Problem of Meaning in the Philosophy of Mind*, Cambridge, Mass., M.I.T. Press.
 - Glass, A., Holyoak, K., and Santa, J. (1979), *Cognition*, Reading, Mass., Addison-Wesley.
 - Kitcher, P. (1984), '1953 and All That: A Tale of Two Sciences', *Philosophical Review* 93, pp.335-373.
 - Marr, D. (1982), *Vision: A Computational Approach*, San Francisco, Freeman & Co.
 - McClamrock, R. (in press), 'Methodological Individualism Considered as a Constitutive Principle of Scientific Inquiry', forthcoming in *Philosophical Psychology*.
 - McClelland, J. and Rumelhart, D. (1985), 'Levels Indeed! A Response to Broadbent', *Journal of Experimental Psychology: General* 114, pp.189-192.
 - Oatley, K. (1980), *Perception and Representation*, New York, Free Press.
 - Pylyshyn, Z. (1984), *Computation and Cognition*, Cambridge, Mass., M.I.T. Press.
 - Simon, H. (1981), 'The Architecture of Complexity', In H. Simon, *The Sciences of the Artificial*, Cambridge, Mass., M.I.T. Press, pp.193-229.
 - Smolensky, P. (1988), 'On the Proper Treatment of Connectionism', *Behavioral and Brain Sciences* 11, pp.1-23.
-