388

1

In H.H. Pattee (Ed.),
Hierarchy theory. New York:
G. Brazileer, 1973. Also in
Models of discovery, Chapter
4.4, H.A. Simon, Reidel Pub.

The Organization of Complex Systems

HERBERT A. SIMON

THE Nobel Laureate Hideki Yukawa earned his prize by observing that the neutron and the proton required a strong, localized force field to hold them together in the atomic nucleus, and that this field should have the properties of a particle—the particle we now know as the pi-meson or pion. The organizers of this series of lectures, having described it as "an experiment in communication between physicists and biologists," evidently concluded that those two kinds of particles—physicists and biologists—also required a binding force to hold them in stable communication. Borrowing Yukawa's idea, they invited me—a behavioral scientist by training—to serve as the pion for the series.

Although I am neither physicist nor biologist, I cannot claim complete innocence of the subject of complex, hierarchic systems, for human administrative organizations—business firms, governments, universities, churches—to which I have devoted a considerable part of my research, are excellent examples of such systems. Since human organizations are staffed by human beings, and since human beings are biological organisms, it might be argued that my research problem is indeed biological. And since biological organisms are constructed from molecules, and those molecules from atoms, and the atoms from elementary particles—all obeying the laws of quantum mechanics—it might even be argued that research on human organizations is merely a rather baroque branch of physics.

I do not intend, however, to talk specifically about either biology or physics. The main point of my paper will be that there are properties common to a very broad class of complex systems, independently of whether those systems are physical, chemical, biological, social, or artificial. The existence of these commonalities is a matter of empirical observation; their explanation is, in a broad sense, Darwinian—they concern properties that facilitate the evolution and survival of complexity. I will leave to the other speakers in the series the specific applications of a general theory of complexity to biological phenomena.

My remarks will fall under four main headings. First, I will define what I—and I hope the other speakers in the series—mean by "hierarchy." Second, I will review briefly two extant pieces of mathematical theory about hierarchies: One has to do with the time required for their evolution, the other with the interaction of their parts. Third, I will explore some consequences of the fact that biological organisms have hierarchic structure. Fourth, I will draw implications from the hierarchies of nature for the hierarchy of the sciences.

In dealing with each topic, I will turn to two main sources of illustration and example, making my discourse into a sort of two-part fugue. On the one hand, I will draw examples from biology, and occasionally from chemistry and physics. On the other hand, I will draw examples from computer science, and specifically from the structure of computer programming languages and programs. I hope that the relation between these two sets of examples will become clear as I proceed.

Hierarchy

In discussions of the theory of complex systems, the term "hierarchy" has taken on a somewhat generalized meaning, divorced from its original denotation in human organizations of a vertical authority structure. In application to the architecture of complex systems, "hierarchy" simply means a set of Chinese boxes of a particular kind. A set of Chinese boxes usually consists of a box enclosing a second box, which, in turn, encloses a third—the recursion continuing as long as the patience of the craftsman holds out.

The Chinese boxes called "hierarchies" are a variant of that pattern. Opening any given box in a hierarchy discloses not just one new box within, but a whole small set of boxes; and opening any one of these component boxes discloses a new set in turn. While the ordinary set of Chinese boxes is a sequence, or complete ordering, of the component boxes, a hierarchy is a partial ordering—specifically, a tree.

It is a commonplace observation that nature loves hierarchies. Most of the complex systems that occur in nature find their place in one or more of four intertwined hierarchic sequences. One partial ordering of boxes starts with observable chemical substances. Analysis of these discloses sets of component molecules. Within the molecules are found atoms, within the atoms, nuclei and electrons, and finally—or is it momentarily?—within the nuclei are found elementary particles.

A second important hierarchy runs from living organisms to tissues and organs, to cells, to macromolecules, to organic compounds, to a junction with the molecules of the first hierarchy. A third, intertwined hierarchy leads from the statistics of inheritance to genes and chromosomes, to DNA, and all that.

A fourth hierarchy, not yet firmly connected with the others, leads from human societies to organizations, to small groups, to individual human beings, to cognitive programs in the central nervous system, to elementary information processes—where the junctions with the tissues and organs of neurobiology largely remain to be discovered.

In this fourth hierarchy, I have included components called "programs" and other components called "elementary information processes." Walter Pitts once referred to this system as "the hierarchy of final causes called the mind." Until about twenty-five years ago, programs and elementary information processes were to be found only as components of biological organisms. Since that time, programs and elementary information processes have been occurring with growing abundance in the artificial complex systems called digital computers. Since programs are much more readily accessible to study in their artificial than in their natural environments, we have learned enormously more about them in our generation than in all previous history. For this reason, the digital computer is taking its place alongside Drosophila, Neurospora, and bacteriophage as an experimental system of the greatest importance. It is for this reason, also, that I shall parallel my biological examples with examples drawn from computer science.

Some Theory of Hierarchy

Several theoretical results are available today on the general behavior of hierarchic systems. I wish to mention two: One providing some explanation for the frequent occurrence of hierarchies in nature, the other showing that there are certain general properties that all hierarchic systems can be expected to possess, wherever they fit in the ordering of

Chinese boxes, and whatever they are made of. I will review these two results here only briefly because I have previously treated them at some length in an essay recently reissued as the fourth chapter of my Compton Lectures at MIT, entitled "The Sciences Of The Artificial" (Cambridge, Massachusetts, MIT Press, 1969).

THE SPEED OF EVOLUTION

One can show on quite simple and general grounds that the time required for a complex system, containing *k* elementary components, say, to evolve by processes of natural selection from those components is very much shorter if the system is itself comprised of one or more layers of stable component subsystems than if its elementary parts are its only stable components. The mathematics of the matter is a straightforward exercise in probabilities, but the gist of it can be given even more simply in a parable.

Two watchmakers assemble fine watches, each watch containing ten thousand parts. Each watchmaker is interrupted frequently to answer the phone. The first has organized his total assembly operation into a sequence of subassemblies; each subassembly is a stable arrangement of 100 elements, and each watch, a stable arrangement of 100 subassemblies. The second watchmaker has developed no such organization. The average interval between phone interruptions is a time long enough to assemble about 150 elements. An interruption causes any set of elements that does not yet form a stable system to fall apart completely. By the time he has answered about eleven phone calls, the first watchmaker will usually have finished assembling a watch. The second watchmaker will almost never succeed in assembling one—he will suffer the fate of Sisyphus: As often as he rolls the rock up the hill, it will roll down again.

It has been argued on information-theoretic grounds—or, what amounts to the same thing, on thermodynamic grounds —that organisms are highly improbable arrangements of matter; so improbable, in fact, that there has hardly been time enough, since the Earth's creation, for them to evolve. The calculation on which this argument is based does not take account of the hierarchic arrangement of stable subassemblies in the organisms that have actually evolved. It has erroneously used the analogy of the second, unsuccessful watchmaker; and when the first watchmaker is substituted for him, the times required are reduced to much more plausible magnitudes.

Specifically, on the simplest assumptions, the mathematical model shows that if a system of *k* elementary components is built up in a many-level hierarchy, and *s* components, on the average, combine at any level into a component at the next higher level, then the expected time of evolution for the whole system will be proportional to the logarithm to base *s* of *k*. In such a hierarchy, the time required for systems containing, say, 10^{25} atoms to evolve from systems containing 10^{23} atoms would be the same as the time required for systems containing 10^{3} atoms to evolve from systems containing 10 atoms. The form of the generalization is interesting, in that it describes a relation between two levels of a system that is independent of absolute level.

We conclude that hierarchies will evolve much more rapidly from elementary constituents than will non-hierarchic systems containing the same number of elements. Hence, almost all the very large systems will have hierarchic organization. And this is what we do, in fact, observe in nature.

NEAR-DECOMPOSABILITY

Most interactions that occur in nature, between systems of all kinds, decrease in strength with distance. Hence, any given "particle" has most of its strong interactions with nearby particles. As a result, a system is likely to behave either as made up of a collection of localized subsystems or to form a more or less uniform "tissue" of equally strong interactions. An example of the former would be a system of molecules; an example of the latter would be a crystal or a piece of rock. Systems of the former kind are, again, hierarchies.

Thus, protons and neutrons of the atomic nucleus interact strongly through the pion fields, which dispose of energies of some 140 million electron volts each. The covalent bonds that hold molecules together, on the other hand, involve energies only on the order of 5 electron volts. And the bonds that account for the tertiary structure of large macromolecules, hence for their biological activity, involve energies another order of magnitude smaller—around one-half of an electron volt. It is precisely this sharp gradation in bond strengths at successive levels that causes the system to appear hierarchic and to behave so. As Melvin Calvin has put it: "This is one of the fundamental things we have to teach freshmen: What is the difference between an atom and a molecule? An atom interacts at one energy level and molecules interact at the other, and that is how we tell the difference." (See Diane M. Ramsey, ed. 1967)

Suppose we were to write down and solve the equations describing the behavior of a hierarchic system having *n* degrees of freedom. We would obtain *n* frequencies, not necessarily all distinct, in terms of which we could describe the dynamic behavior. We could arrange these frequencies in a sequence, beginning with the lowest frequencies corresponding to the slowest oscillations, and going down through medium-range frequencies, to very high frequencies at the end of the list. As is well known, in the case of the physical system I described a moment ago—a system of macromolecules—Planck's Law prescribes a strict proportionality between bond energies and the associated frequencies.

If we now observe the behavior of the system over a total time span, T, and our observational techniques do not allow us to detect changes during time intervals shorter than τ, we can break the sequence of characteristic frequencies into three parts: (1) low frequencies, much less than 1/T; (2) middle-range frequencies; and (3) high frequencies, greater than $1/\tau$. Motions of the system determined by the low-frequency modes will be so slow that we will not observe them—they will be replaced by constants.

Motions of the system determined by the high frequency modes will control, for the reasons already given, the internal interactions of the components of the lower level subsystems in the hierarchy, but will not be involved in the interactions among those subsystems. Moreover, these motions will be so rapid that the corresponding subsystems will appear always to be in equilibrium and most of their internal degrees of freedom will vanish. In their relations with each other, the several subsystems will behave like rigid bodies, so to speak.

The middle band of frequencies, which remains after we have eliminated the very high and very low frequencies, will determine the observable dynamics of the system under study—the dynamics of interaction of the major subsystems. As we have seen, these dynamics will be nearly independent of the detail of the internal structure of the subsystems, which will never be observed far from equilibrium. Hence, we can build a theory of the system at the level of dynamics that is observable, in ignorance of the detailed structure or dynamics at the next level down, and ignore the very slow interactions at the next level up. The goodness of our approximation will depend only on the sharpness of the separation of the high frequencies from the middle-range frequencies, and of the middle-range frequencies from the low frequencies. We will, of course, want to select the boundaries so as to make that separation as sharp as possible. I will have a little more to say about the relation between the layered structure of natural phenomena and the layered structure of theories.

Systems with the sorts of dynamic properties that I have just described are called "nearly-decomposable" or sometimes "nearly completely decomposable" systems. A rigorous mathematical basis exists for reaching the conclusions I have just stated about such systems, but I think our intuitions will suffice for present purposes. (See Ando et al., 1963.)

Hierarchies in Computing Systems

So far I have used natural systems as my examples of hierarchies. I could as well have used modern computing systems and their behavior. I should now like to describe a computer program called EPAM with which I am familiar. EPAM simulates human laboratory subjects in certain simple learning tasks, but just what it does is of no concern to us here. What is important is that it is a large, complex computer program.

EPAM consists of lists of instructions organized as "routines." It is written in a computer programming language called IPL-V, to which I will return in a moment. The instructions—there are about 3,000—are of two kinds: (1) primitive instructions, corresponding to a fixed, basic set of IPL-V instructions, and (2) higher-level instructions. Whenever I write an IPL-V routine consisting of a list of primitive instructions, I can give that routine a name. I can then use that name just as though it were an instruction—a higher-level instruction—in any other routine I wish to write. Whenever the system, during execution, encounters such a higher level instruction, it simply executes the subroutine that the instruction names. There is no limit to the allowable number of levels of subroutines, and at various points EPAM is five or even ten levels deep.

But that is not all. The IPL-V primitives, in terms of which the EPAM routines are ultimately defined, are themselves not very primitive. They correspond, in fact, to routines—some of them fairly complex—written in the instruction language of the particular kind of computer on which EPAM is to be run. For each distinct machine there must be a translation of IPL-V into the language of that machine; but the behavior of EPAM is substantially independent of that translation and indifferent to what machine it is run on. We can say that EPAM has a "meaning" that is independent of the particular machine language in which it is expressed.

We are still far from having probed the bottom levels of our hierarchy. Having reached the level of machine instructions, we can analyse how these instructions are realized in the logical organization of the computer itself. The study of that logical organization leads, in turn, to lower hierarchic levels, where we first encounter the actual physical devices that implement the behavior and the actual physical laws that govern those devices. Just as the same language—for example, IPL-V—can be implemented on vastly different computers (more than a dozen translations exist today), so the same computer design, at the logical level, can be implemented with entirely different hardware devices. From a programming standpoint, the IBM 709 and the IBM 7090 were almost identical machines, although the former made extensive use of vacuum tubes, while the latter was a solid-state system. From a physical standpoint, they were radically different machines.

The system I have described is a nearly-decomposable system. Its highest frequencies are those associated with the physical components of the computer—nowadays, microsecond or nanosecond frequencies. Frequencies associated with the logical organization of the machine and its machine instructions might be, for a fairly fast machine, say, in the range of ten microseconds. IPL-V instructions are executed at millisecond rates (one or two orders of magnitude slower than machine instructions). Some of the higher-level routines of EPAM take seconds to execute—even on a fast modern computer, EPAM requires several seconds to memorize a nonsense syllable.

Now just as we can reach an approximate understanding of a physical system at the level of chemical reactions, ignoring what is going on within the atoms and nuclei, so we can reach an approximate understanding of EPAM by considering only a few of the highest levels of the program, not going down even to IPL-V primitives, much less to machine language, logical design, or computer physics. As a matter of fact, since IPL-V translators are explicitly constructed to make behavior machine-independent, we should be able to describe EPAM *exactly* (apart from some speed parameters) in terms of IPL-V. How much accuracy we lose in disregarding all but the high-level routines depends on how carefully we have sealed off each level from those below.

What do I mean by "sealing off"? Each year when I fill out my income tax form, I am instructed to perform certain additions and subtractions, and even a few multiplications. I am told where to find the operands and where to enter the sum, difference, or product. Later, the Internal Revenue Service audits my return, and if I have made a mistake—as I sometimes do—corrects it. So the IRS can tell whether I have done the arithmetic correctly, but it cannot tell *how* I did it—what subroutine I use to define "acdition" or "multiplication." Perhaps I multiply with paper and pencil, from right to left, or from left to right; perhaps I do it in my head, or on a desk calculator, or on my university's computer; perhaps my wife does it. The only communication between my arithmetic routines and the IRS's auditing routines is through the inputs and outputs of my processes; the processes themselves are immune from scrutiny.

When I multiply two four-digit numbers together, I have to keep the multiplier and multiplicand in memory or on paper. Then I have to store temporarily the four partial products—four or five digits each. When I am done, I have to retain only the seven or eight digits of the final product. The unobservability (to the IRS) of the intermediate results that I create, but hold only temporarily, is precisely analogous to the unobservability of the high-frequency dynamics of a hierarchic system, the disequilibrium dynamics of the smallest components. All of this detail is simply irrelevant to the lower-frequency interactions among the larger segments. No matter which of several processes I use to obtain the product, and which intermediate results I obtain *en route,* the final information I obtain and pass on to other routines is the same. Hence, hierarchy is associated with a very fundamental form of parsimony of interactions. The art of subroutining, in writing complex computer programs, consists in discovering the points of cleavage at which the least information needs to be passed from one subroutine to another.

LOOSE HORIZONTAL COUPLING

In describing the behavior of nearly-decomposable systems, I emphasized "vertical" separation—the segregation of the low-frequency from the high-frequency dynamics. The last examples suggest that the theory of near-decomposability can be extended to say something about the horizontal relations among subsystems at the *same* hierarchic level.

Consider, again, the frequencies of a nearly-decomposable system arranged in order from low to high. We now observe the behavior of the system much more microscopically than we did before, so that we need consider only the roots of frequency greater than $1/\tau$. This is equivalent to ignoring the weak interactions among the subsystems of the nearly-decomposable system and treating the subsystems as completely decoupled from one another. But then we can take the remaining high-frequency roots and assign them to their respective subsystems. Particular frequencies describe the behavior of particular subsystems.

Returning to our original system, we see that the frequencies describing its dynamics can be *partially* ordered, and each subset of frequencies in the partial ordering (formally, an equivalence class at some particular level of the ordering) can be associated with a specific subsystem in the partial ordering of system components. There will be, essentially, an isomorphism between the hierarchy of subsystems and the hierarchy of equivalence classes of frequencies describing the system, and particular frequencies will "belong" to particular subsystems.

To a first approximation, the behavior of any given subsystem will depend only on the frequencies belonging to it, together with the lower frequencies belonging to systems at higher levels of the hierarchy. It will be independent of the frequencies associated with other subsystems at the same or lower levels of the hierarchy. (I am sorry that high "frequencies" correspond to low "levels," but it can't be helped.)

The loose horizontal coupling of the components of hierarchic systems has great importance for evolutionary processes just as the loose vertical coupling does. The loose vertical coupling permits the stable subassemblies to be treated as simple givens, whose dynamic behavior is irrelevant to assembling the larger structures, only their equilibrium properties affecting system behavior at the higher levels.

The loose horizontal coupling permits each subassembly to operate dynamically in independence of the detail of the others; only the inputs it requires and the outputs it produces are relevant for the larger aspects of system behavior. In programming terms, it is permissible to improve the system by modifying any one of the subroutines, provided that the subroutine's inputs and outputs are not altered.

When the same outputs can be obtained from the same inputs by two or more different paths, we usually speak of "functional equivalence." Functional equivalence, of course, is not peculiar to computer programs, for it occurs frequently in natural phenomena. In chemical reactions, for example, isotopic variants of atoms of the elements are usually functionally equivalent—as long as two atoms present to the surrounding environment the same configuration of outer-shell electrons and differ only slightly in atomic weight, their chemical behaviors are almost indistinguishable.

In biological systems, innumerable examples of functional equivalence are provided by multiple reaction pathways. The equivalence can refer to the reaction itself—for example, the two pathways for synthesis of lysine, one employed by some fungi and euglenids, the others by most plants. Alternatively, the equivalence can refer to the enzymic apparatus controlling the reaction—for example, the wide variety of chemically distinguishable protein molecules that serve as functionally equivalent hemoglobins, both among different species and even in a single species.

The various functional equivalents may, of course, vary widely in their metabolic efficiency, and their relative efficiencies may depend on environmental circumstances as well—horse hemoglobin seems to work better for horses and human hemoglobin for people, although perhaps that is only for immunological reasons. But, of course, it is precisely because they may vary in efficiency that functional equivalents have significance for natural selection. Functional equivalence permits mutation and natural selection to go on in particular subsystems without requiring synchronous changes in all the other systems that make up the total organism.

The loose horizontal coupling of components can be observed at all levels of hierarchic structures. Thus, a mammal's circulatory system is loosely coupled to other systems. It receives oxygen from the respiratory system and nutrients from the digestive system. It delivers these to the muscles, say, from which it receives carbon dioxide, and other wastes. These it delivers, in turn, to lungs and kidneys, and so on.

Just how the circulatory system accomplishes these tasks is of no concern, so to speak, to the other systems, as long as it does accomplish them. Appropriate evolutionary changes may take place in any one of these systems without necessarily, or immediately, disturbing the others. Natural selection may improve a horse's locomotion without necessarily changing his digestion, although changes in the metabolic rates associated with one subsystem may, on a longer time scale, bring about natural selection of new adaptations of other subsystems.

The same kind of argument as that used to show that nearly-decomposable systems will evolve more rapidly than others can be used to demonstrate that the advantage in adaptation will be increased if different components of the organism are coupled to different components of the environment. The point is most easily shown by analogy with problem solving efficiency.

Consider the problem of cracking a safe that has 10 dials, each with 10 possible settings. To find the combination by trial and error would require, on the average, testing half the total number of possible settings—half of 10^{10}, or 5 billion. If each dial emits a faint click when it is set to the correct number, the safe-cracking job becomes trivially simple. Now, on average, only $5 \times 10 = 50$ settings will have to be tried.

PRODUCTION SYSTEMS

The loose horizontal coupling of subsystems can be exploited in another way: to make each subsystem independent of the exact timing of the operation of the others. If subsystem B depends upon subsystem A only for a certain substance, then B can be made independent of fluctuations in A's production by maintaining a buffer inventory of the substance upon which B can draw. The storage of fat is a well-known and important biological example of this principle. Buffer inventories permit many interdependent processes to operate in parallel, at fluctuating rates and under only feedback control—as in the familiar mechanism by which the inventory of the substance produced by the last enzyme in a chain inhibits the activity of the first enzyme.

Most digital computers are organized, more or less, as serial one-process-at-a-time devices. The idea of loosely coupling their processes in the way just described can be employed to simulate parallel systems. To do this each routine is written as a "production" in two parts: The first part tests for the presence or absence of certain conditions; if and only if the conditions are satisfied, the second part carries out its characteristic process. Clearly, there is a close logical relation between such productions and the operons of molecular genetics. If one wanted to write a computer simulation of operons, one would represent them by productions. Because their components are so loosely coupled, production systems are much more easily modified, component by component, than are computer programs written as more traditional subroutine structures. As a result, complex programs are increasingly being organized in this form.

Alphabets

The flexibility of coupling among subsystems can be further enhanced by limiting the variety of different kinds of components that are incorporated in the larger systems. When the numerous component elements (called "tokens") of the subsystems of a hierarchy all belong to a small number of basic types, we call this set of types an "alphabet." A common milk protein contains 5,941 "tokens"—atoms. All of these atoms belong to the five elements—types C, H, O, N, and S. These five types are drawn from the 92-letter alphabet of natural elements.

The alphabet of primitive instructions in IPL-V is more baroque than the atomic alphabet. It contains about 150 instructions, but if we treat certain similar instructions as isotopes, the number remaining is not far from the number of elements. As Turing and others have shown, a computing system—even a completely general one—can get along with a far smaller alphabet than that. In fact, about five instructions like "write," "erase," "move left," "movᵉ right," and "test and branch" wil! suffice for complete generality. It is convenient and efficient, but not logically necessary, for computer instruction codes to contain more operations than these.

Two alphabets have supreme importance for biology: The alphabet of twenty-odd amino acids, and the alphabet of four (or five) nucleic acids. I will confine my remarks largely to the former, for we know today how the one can be translated into the other.

ALPHABETS, LANGUAGES, AND PROGRAMS

Not every level in a hierarchic structure is characterized by a small alphabet of components. There are only 92 natural elements, but innumerable molecules at the next level up; there are only about 20 amino acids, but innumerable protein molecules. There are only 150 primitive IPL-V instructions, but innumerable routines written in terms of them—at least thousands. What significance can we attach to the fact that only certain hierarchic levels are alphabetic?

We must distinguish between alphabets and languages, on the one hand, and programs or messages, on the other. Alphabets and languages are systems that provide a potentiality for communicating any of a large number of programs or messages. They consist of elements, and rules for the combination of the elements into messages. We may regard alphabets simply as those languages that are based on small numbers of different elements (as distinct, for example, from natural languages, which typically contain hundreds of thousands of morphemes or words).

Members of a single organization may share a set of common messages—standard operating procedures, say. Interaction throughout a language community takes place by means of a common language, messages being constructed and transmitted as needed. Alphabets, because of their restricted set of elements, are even shared across the boundaries of language communities. Most of the Western European languages use the Roman alphabet.

If we knew in advance just what messages were to be sent, we could always find a special encoding that would be more efficient than constructing the messages from a general-purpose language. If we knew in advance the subject of the messages, a lexicon could provide a more efficient encoding of messages than is provided by the combinations of a small alphabet. The "inventory" of elements we would have to keep on hand would be much greater, however, for the lexicon than for the alphabet.

To realize its potential advantages for communication, a language should have these characteristics: (1) sufficient variety in its primitive processes so that no meaning is absolutely excluded from expression, and (2) sufficient flexibility in its rules of combination so that any nuance can be expressed by building up composite structures. What is required of the amino acids, and of the nucleic acids, is that they provide sufficient variety so that their combinations, proteins, and chromosomes, respectively, can perform all of the basic kinds of chemical functions that need to be performed in the cell.

This does not explain, however, why the nucleic acid and amino acid languages are based on alphabets. If this characteristic has significance for evolutionary success, the significance appears to be different in the two cases. What is needed in the genetic case is a simple code that is isomorphic to the set of amino acids—hence nothing is to be gained from a large alphabet. But what about the amino acids themselves?

An organism can only survive in an environment containing appropriate nutrient matter. Unless it can control that environment, or unless the environment is controlled beneficiently by a higher-level system, it cannot rely on finding in the environment highly particular substances. We would expect alphabetic languages to be prominent in communication at subsystem boundaries where each subsystem experiences considerable uncertainty as to what it will find in its environment—where it cannot count on the environment to provide inputs tailored to its exact needs.

(I may observe that manufacturing concerns behave in exactly the same way. They tend to hold their in-process inventories in the form of generalized intermediate products that are capable of being formed into a variety of final products—ingots rather than special steel shapes, for example.)

It is hardly surprising, therefore, that the transactions of an organism with its environment (and even remote internal transactions via its circulatory system) are handled with an amino acid currency, and not with a protein currency. Proteins are far too specific in function, and far too closely adapted to a particular type of organism, to be exchanged satisfactorily among organisms. An amino acid molecule in the bloodstream of an animal may have come from many different sources. It may have been obtained by digestion of protein foods of various kinds; it may have been synthesized from other amino acids; it may have been hydrolyzed from proteins in the animal's own tissues. Two molecules of the same amino acid are functionally equivalent, however derived.

An organism will have access to a supply of components if it maintains itself in a broth of potential replacement parts. It would be hard-pressed—at least without cannibalism—to find such a broth of appropriate proteins.

SUMMARY: LOOSE COUPLING

Our whole discussion to this point underscores the crucial significance of hierarchic organization to the synthesis and survival of large, complex systems. To a Platonic mind, everything in the world is connected with everything else—and perhaps it is. Everything is connected, but some things are more connected than others. The world is a large matrix of interactions in which most of the entries are very close to zero, and in which, by ordering those entries according to their orders of magnitude, a distinct hierarchic structure can be discerned.

By virtue of hierarchic structure, the functional efficacy of the higher-level structures, their stability, can be made relatively independent of the detail of their microscopic components. By virtue of hierarchy, the several components on any given level can preserve a measure of independence to adapt to their special aspects of the environment without destroying their usefulness to the system.

Reduction

I will close with some remarks about reductionism and the structure of the sciences. The general tenor of these remarks should now be predictable. There are at least two versions of the concept of explanation in science. In both versions, of course, explaining a phenomenon involves reducing it to other phenomena that are, in some sense, more fundamental.

But with agreement on this point, the two concepts of explanation branch. The one concept—let me call it Laplacian—takes as its ideal the formulation of a single set of equations describing behavior at the most microscopic, the most fundamental level, from which all macrophenomena are to follow and to be deduced. No one, of course, believes that the program could actually be carried out—the equations, when written, would be far too hard to solve. In spite of that, the concept has practical consequences in the real world, for it influences some scientists' choices of research problems—their view of what is "really" fundamental.

The second concept—for lack of a better name let me call it Mendelian—takes as its ideal the formulation of laws that express the invariant relations between successive levels of hierarchic structures. It aims at discovering as many bodies of scientific law as there are pairs of successive levels—a theory of particle physics, one of atomic physics, one of molecular chemistry, one of biochemistry, and so on. Since the world of nature is a nearly-decomposable system, and since the invariant properties of a nearly-decomposable system have this layered quality, the fundamental scientific laws must take this form also.

Since, in the second view, nature is only *nearly*-decomposable, not *completely* decomposable, many of the most beautiful regularities of nature will only be approximate regularities. They will fall short of exactness because the properties of the lower-level, higher-frequency subsystems will "show through" faintly into the behavior of the higher-level, lower-frequency systems. Thus, for example, there is a fundamental truth in Prout's hypothesis—that the atomic weights of all the elements can be expressed as integers—even though we know it is not an exact truth, and know the relativistic explanation for the mass deficiency. We know the vicissitudes that Prout's hypothesis suffered: How it was discredited by the 19th century's measurement, with continually increasing accuracy, of fractional atomic weights; how it was triumphantly vindicated by the discovery of isotopes; how further increases in the accuracy of measurement put it in doubt again.

If we were to make a list of the most important, the most beautiful laws of natural science that have been discovered in the past three centuries, we would see that the vast majority of them hold only approximately, and only if we are willing to ignore details of microstructure. The pattern expressed by these laws is simply not present in the underlying, detailed Laplacian equations.

I do not want to present a one-sided case. The fact that nature is hierarchic does not mean that phenomena at several levels cannot, even in the Mendelian view, have common mechanisms. Relativistic quantum mechanics has had spectacular success in dealing with phenomena ranging all the way from the level of the atomic nucleus to the level of tertiary structure in organic molecules.

Perhaps a balanced way to state the matter is this: Suppose you decided that you wanted to understand the mysterious EPAM program that I have, without explaining, alluded to several times in this paper. I could provide you with two versions of it. One would be the IPL-V version—the form in which it was actually written—with its whole structure of routines and subroutines. If you were curious about its implementation on a computer, I could supplement the EPAM program with a listing of the program that translates IPL-V ins:.uctions into machine-language instructions for some particular machine.

Alternatively, I could provide you with a machine-language version of EPAM after the whole translation had been carried out—after it had been flattened, so to speak, and spread out in all its Laplacian detail. I don't think I need argue at length which of these two versions would provide the most parsimonious, the most meaningful, the most lawful description of EPAM. I will not even propose to you the third, and truly Laplacian possibility—of providing you with neither program, but instead, with the electromagnetic equations and boundary conditions that the computer, viewed as a physical system, would have to obey while behaving as EPAM. That would be the acme of reduction and incomprehensibility.

Notice that in my plea for a hybrid Laplacian-Mendelian approach to fundamental science I have given no defense of vitalism, nor have I alluded to the Heisenberg Uncertainty Principle. Both of these seem to me red herrings across our particular path of inquiry. Scientific knowledge is organized in levels, not because reduction in principle is impossible, but because nature is organized in levels, and the pattern at each level is most clearly discerned by abstracting from the detail of the levels far below. (The pattern of a halftone does not become clearer when we magnify it so that the individual spots of ink become visible.) And nature is organized in levels because hierarchic structures—systems of Chinese boxes—provide the most viable form for any system of even moderate complexity. I have tried in this paper to show some of the deeper reasons why this is so.

REFERENCES

Ando, A., Fisher, F. M., and Simon, H. A. 1963. *Essays on the Structure of Social Science Models.* Cambridge, Massachusetts: MIT Press.

Ramsey, Diane M., ed. 1967. *Molecular Coding Problems*, pp. 120–121. New York: New York Academy of Sciences.