DAVID KLAHR and *HAROLD J. LEAVITT*

*Tasks, Organization Structures, and Computer Programs**

As WE TRY to assess the present state of the relationship between computers and organizations, the following general phenomena seem to stand out.

1. Computers entered industry as massive, fast, brute force arithmetic devices. They were bought because managers believed they might do known work more rapidly, accurately, and cheaply than organized sets of people. This was the EDP phase of computerization. Its greatest effect on organizations was chiefly just what was intended: a substitution of computer programs for human substructures that were then doing the same routine tasks.

2. Starting about 1955, an operations research, "problem-

* This research was supported in part by Ford Foundation Grant 1–4005 to the Carnegie Institute of Technology for research on organizational behavior, and in part by a Ford Foundation Fellowship to Klahr.

solving" phase of computerization got underway. This time its organizational impact was upon less highly routinized substructures, like inventory control or production scheduling groups. And the effect was partially or completely to program some tasks that had either been performed by judgment in the past, or had not been performed at all.

3. More recently we have heard much and seen a little of an "information systems" wave of computerization; this time characterized by a marriage between computer programs and communications technology. Presumably the organizational impact of this wave may be broader and may move into somewhat higher levels of the organizational pyramid. One effect of this new thrust seems to be the binding together of several subparts of the total structure, often severely modifying some in the process. Another is to move the locus of large amounts of information to some central point; and, in some cases, to change the locus of certain decisions from one part of the structure to another. The still speculative part of real-time information systems forecasts the provision of instantaneous information to top executives at their will — though no one is quite sure what they should will, or how such innovations will affect present structures.

4. In the academic background, but not yet much apparent in industry, lie heuristic computer programs — programs that reduce the solution space of problems to a manageable size by making good, but not infallible, guesses. These programs do very complex things; things which, when done by humans, we call thinking or learning or adapting.

If we consider the present organizational effects of these four waves together, it seems clear that top managers' jobs have not been much affected by computers, although many lower-level routine jobs have. Effects on intermediate level people and structures are not clear. There is an increasing amount of decision-making using computers and systems, but certainly middle line managers have not been swept away.

And although centralization of information seems to be coming, centralization of decisions is not so apparent. Central information storehouses are sometimes being used for centralized decision-making, and sometimes as pools into which people at all levels may dip to help them make the same or more decisions than they made before.

But these effects are largely attributable to the EDP–OR uses of computers. The difficult issue is to assess the impending effects of information systems and heuristic programming.

As we view it, we have been sitting for several years on an EDP–OR plateau,[1] albeit an expanding one. We think we may just be taking off on a new spurt of information systems, and that in a few years a heuristic spurt will follow.

But nothing is to be gained by trying to justify that belief or to reassert (defensively on the part of one author) that 1966 is *not* 1986.[2] Instead, we shall use this opportunity to try to conceptualize — hopefully for others as well as ourselves — the relationships among (1) organizational tasks, (2) organizational structures, and (3) computer programs. We have a few years of perspective now, so perhaps we can now do more than speculate. Perhaps we can now try to relate and model,

[1] The extent of this plateau is best illustrated by noting the similarity of our general predictions to some made over eight years ago: "Operations research has made large contributions to those management decisions that can be reduced to systematic computational routines. To date, comparable progress has not been made in applying scientific techniques to the judgmental decisions that cannot be so reduced. Research of the past three years into the nature of complex information processes in general, and human judgmental or heuristic processes in particular, is about to change this state of affairs radically. We are now poised for a great advance that will bring the digital computer and the tools of mathematics and the behavioral sciences to bear on the very core of managerial activity — on the exercise of judgment and intuition; on the process of making complex decisions." H. A. Simon and A. Newell, "Heuristic Problem Solving: The Next Advance in Operations Research," *Operations Research,* 1958, *6,* No. 1, 1–10.

[2] For another example of predictions that extrapolated the rapid development of the 1950's into 1980's, see H. J. Leavitt and T. L. Whisler, "Management in the 1980's," *Harvard Business Review,* 1958, *36,* No. 6, 41–48.

using as our base what has happened in the last decade to both organizations and computer programs.

I. INVARIANCE BETWEEN TASKS AND STRUCTURES

We shall make the following sweeping assertion: In the long run, and "other things equal," men will build similar organizational structures to perform similar tasks. Armies are more like one another than they are like universities. Universities are more like one another than they are like armies.

By organizational structures we mean what is usually meant: a set of roles and role relationships, a communication structure, and a set of ordered work flows. By task we also mean what is usually meant: a roughly defined, generally perceived and accepted goal.

Consider, for example, at some time before the advent of computers, the task of preparing a payroll in a large manufacturing company. This recurs regularly and is large enough to require the activities of more than one person, simply because the computations are numerous, and the time constraints tight.

Over time, the organization develops a structure that can cope with the task. The operational steps in preparing the payroll are specified; individuals are assigned to perform particular operations in particular sequences. The task becomes programmed.

It is highly unlikely, given such a task, that humans could fail to program it into a structured and orderly form. Since human beings are reasonably flexible tools, competent both to perform the required operations and the necessary coordinating and supervising activities, and since the task is large enough to require more than one human being, the outcome over the long run will be some variety of specialized and hierarchical structure in which work steps are prescribed and assigned to particular roles and in which communication channels are more or less specified.

But perhaps the payroll task is a bad example. It is a simple, well-defined (though large), deadlined, repetitious task. Consider another alternative: the scheduling of production in a factory. That task is also deadlined, though perhaps less tightly. It is repetitious but probably is so over a longer time cycle. Its output is a *plan* rather than a large number of discrete decisions. The proportion of coordinative functions to "doing" functions is much greater. Such a task, we submit, will tend to generate a human organization structure that is less precisely defined, in which relationships are less clearly specified; a structure which is generally more flexible, in which roles are less specialized. Such a task should also generate group or committee functions in addition to hierarchies of individual functions.

If we extend the same idea to the level of ill-structured, novel tasks — like decisions to expand plant capacity — we shall still assert long-term invariance, but now the structure in question should become even less specialized, even less hierarchical, more open, more flexible, and generally more Jellolike. It is paradoxical, perhaps, to describe such an amoebic mass as a structure; yet the organization structures generated by novel, ill-defined tasks seem best characterized by their unclarity, their "amorphousness," their lack of specialization.

It is doubtless obvious by now that our assertion of invariance is of the grossest sort.[3] We cannot adequately categorize either different kinds of tasks or different kinds of structures except in loose and only semioperational terms. Nevertheless, that assertion is both important to our present thesis and, we

[3] C. Faucheux and K. Mackenzie, "Task Dependency of Organizational Centrality," *Journal of Experimental and Social Psychology* (in press) have recently provided a finer experimental demonstration of this relationship. Using the communication networks apparatus, we find that the data demonstrate clearly that the A tasks (which are of a routine deductive nature) lead towards centralization, while the B tasks (which are nonroutine and have some inferential components) do not lead toward centralization.

submit, sensible. It is important to our argument because we shall try to go on to show that computer programs are often interchangeable with organization structures, and that tasks also generate computer programs. So that for a given task, human structures and computer programs may be highly similar in their design.

One deviation is in order before moving on. If we assume that ill-defined, novel tasks are more characteristically found at the tops of large industrial organizations than the bottoms, it will follow that tops of organizations may be quite interchangeable, and that executive committees will tend to be all alike. On the other hand, lower-level substructures, designed for more particular and well-defined tasks, will be more task-specific and probably less interchangeable, except over a narrow range.

This conception seems consonant with the way in which tasks flow into and through organizations. They often enter at the top in ill-defined, new forms. The top works them over, defines and operationalizes them, and then, if they are to become continuing tasks, *passes them down the hierarchy*, where they are again converted from their now partially operational states into highly defined states, and again passed down to specially created or adapted substructures. Presumably the top, in the interim, has turned its attention to other new, ill-defined issues.

Since (1) top structures are presumably determined by their tasks, and since (2) the relevant set of tasks consists of novel, few of a kind, ill-defined ones, and since (3) all sets of novel, ill-defined tasks are alike in their novelty and uniqueness, *then* top structures should be alike in that they are "specialized" for novel, unspecialized tasks.

Having asserted invariance between human structure and task, we are left with the problem of relating computer programs to both human structures and tasks.

We shall try to show in the next few pages that computer programs are analogous to organization structures, that they are often presently interchangeable with some structures, and that they are potentially interchangeable with a much wider set of structures.

II. THE NATURE OF COMPUTER PROGRAMS

Almost all complex computer programs consist of a set of semi-independent subprograms, usually called *closed routines* (CR), tied together by an *executive program* (EP). *Closed routines* are relatively fixed, prepackaged operational activities. Typically they communicate with other routines and with the EP through a limited number of communication channels. As far as the EP is concerned, the subprograms consist of some functional relation between input and output variables. Any of several functionally equivalent routines could be used. The EP views the subprogram as a henchman who says, "Do this. Tell me when you've done it. Give me the results." The routines, in turn, usually have low vision; they have little knowledge of what they are working on. They do their job on any data that are presented to them. They don't care what they are correlating with what or what standard deviation they are deriving. These routines have access to only a small part of the total information in the system.

They allow the EP to refer to a complex sequence of operations with a single name and a few parameters. These CR's provide the same kinds of advantages over *open routines* that standardization and modularity provides in any large organization structure. The early days of programming (and often the early phases of any new program) saw a wide use of open routines: routines that were an integral part of the main program and were designed to fit the specific task at hand.

Perhaps it is a fair analog to suggest that a complex program with several CR's is like an organization which has several central staff departments — a department of market research, a department of chemical analysis, and so on — to which each of the other parts of the organization can send their problems when market-research operations or chemical-analysis operations need to be done. Open routines, in contrast, are "decentralized." A new one is needed for every group in the organization that needs that kind of job done. So there would be several market-research groups, each one designed semi-independently of the others and each one bound primarily to the attributes of the task at hand.

Complex computer programs are "run" by EP's. (The terminology is not well standardized here; some systems use the terms "executive," "supervisor," "monitor.") EP's may vary in structure from very simple listings of the order in which the subprograms are to be executed to a highly complex set of conditional tests and simultaneous operations. They may include just a few levels of activities or hierarchies many layers deep.

The *computational* requirements for the EP's are often trivial, but EP's must sit on top of the program because they must have good vision over the total system. EP's in programs do many things analogous to what executives do in human organizations. For example, they do the following (and we may read at will, "organization" in lieu of "program"):

1. They look *outside* the program, receiving information from the environment in one form and transmitting it back to the environment in another form. They observe what kind of work has piled up, what needs to be done next, etc.
2. They look *inside* the program and they maintain control over its subprograms.
   (*a*) They *detect,* checking what is done and what still needs to be done.
   (*b*) They *interrupt.* They command one subprogram to stop and another to start.
   (*c*) They *monitor,* making sure that no errors or intolerable conditions have occurred or are imminent.
   (*d*) They *allocate* resources, assigning computer time,

space, and computational facilities to the appropriate routines, and they assign processes to problems.

(*e*) They *coordinate*. They make sure that when subparts of the activity are completed, the results are properly fed to the next stage of the process. They schedule things in appropriate sequences and make sure the subparts are gradually put together into a meaningful entity.

(*f*) They do *housekeeping*. They inspect and clean up their own over-all working areas. They clean out unused areas and make them available for new information. The housekeeping routines initialize and finalize the loose ends that the subprograms may have neglected. They keep the total program in fighting trim.

Notice that we have said nothing about decision-making as such. The whole EP part of the computer program is set up to achieve a decision. Of course, in some computer programs the only thing that can be thought of as an output decision is a large number of outputs of some of the lower level CR's. In those cases, all the EP ends with is the knowledge that the tasks that are parceled out are now done. In such programs, the output may be a large number of payroll checks, rather than a set of scheduling decisions.

III. ORGANIZATION STRUCTURES AND COMPUTER PROGRAMS

Let us return for a moment to the example of the payroll before computers. We suggested that a human program (an organizational structure) would have been designed for it; a structure somewhat characterized by specialization of human effort and a hierarchy of authority and control.

If we now introduce the computer program, we can provide an alternative to that specialized and hierarchical set of human beings. Our computer program's closed routines can perform the arithmetic operations and file-maintenance procedures, and the simple EP can also perform many of the coordinating and supervising steps necessary for getting the payroll checks out. If computer programs had existed when the payroll problem arose for the first time, the computer program, rather than human structure, might well have been assigned the task from the beginning — given cost equivalence or cost advantage.

IV. OBJECTS-PROCESSING AND INFORMATION-PROCESSING

But our frequent analogies between organizational structures and computer programs begin to break down if we push into other kinds of organizational activities. In many human organizations, structures are set up to do things that are not done at all by any known computer programs. Objects get worked upon. Raw material is transformed into finished products. Cars roll off the assembly line. In a general sense, the industrial organizational structure, at least the manufacturing substructure, processes objects as well as information.

One way we distinguish among different kinds of industrial organizations is in the extent to which one or the other of these dominates. An insurance company does very little object processing but a great deal of information-processing. With some effort, most of what it does can be fitted into our model of the computer program as an organizational structure. But the automobile manufacturer cannot do this. On the other hand, in order to process automobiles the auto manufacturer must also process large amounts of information. Moreover, while we have long since developed an elaborate technology to help humans process objects, our information processing technology is largely new and is yet to approach a steady state.

We can push the analogy even further by paraphrasing Norbert Wiener.[4] One can describe man as an information-processing and object-processing system. He has a brain and he has a reflex system of interacting eyes, hands, and muscles. He can do high-energy things with his eye-hand-muscle (EHM) set, and he can control those high-energy things with his brain. For a long time, we have had machines that are highly analogous to man's EHM system, especially the muscle part of it — sometimes better, sometimes worse, sometimes more massive than man's muscles. And we now have information-processing machines that are in many ways functionally equivalent to man's brain.

We can thus make a two-by-two characterization of the possible relationships between man and machine. On one dimension, man can be viewed either as an information processor or as an EHM system. On the other dimension, we can consider information-processing machines or EHM machines.

When information-processing machines interact with men, we are in the realm of man-machine systems. This interaction can occur in two ways: (1) When the information-processing machine interacts with the information-processing man, we are in the man-machine areas that extend and augment human thought.[5] (2) When the information-processing machine interacts with the EHM man, we see the computer-controlled astronaut or the farmer who is told by computers when to sow and when to reap.

The two additional interactions that complete our four-way classification have been with us much longer. (3) Information-processing man interacting with EHM machine covers

[4] Norbert Wiener, *The Human Use of Human Beings,* Boston: Houghton-Mifflin Co., 1950.

[5] Or as M.I.T.'s Project MAC calls it: "Machine Aided Cognition."

the range from the first industrial revolution to the driver–automobile relationship. (4) When EHM man interacts with EHM machines, perhaps we are describing the relationship between man and his early tools and beasts of burden.

V. COMPUTER PROGRAMS AND ORGANIZATION STRUCTURES FOR ILL-DEFINED TASKS: HEURISTIC PROGRAMS AND REAL-TIME SYSTEMS[6]

In this section we shall attempt to extend our discussion of the similarities between organizations and computer programs upward in the organization. We shall offer some assertions about the kinds of programs that may be able to do to the upper reaches of management what has already been done to the lower regions. We shall argue that (1) at the upper levels all human organization structures begin to follow heuristic procedures, (2) all are organized in a very shallow hierarchy, (3) computer programs will take on the same heuristic, shallow-form when extended to this level of managerial decision-making, and (4) real-time information systems will be the primary carriers of computerized decision-making to upper organizational levels.

Earlier we proposed that in the long run, organizational structures were predictable from knowledge of task; and that upper-level structures tended to take a flexible, "organic" form because the tasks at these levels tended to be ill-structured, variable, and novel.

We then argued that top-level organizational structures should tend to look alike even across industries; that is, alike in their "amorphousness." Job descriptions at that level are nonoperational and nonspecific; informal coalitions tend to

[6] The real-time portion of this section was in draft when we received, with pleasure, Donald C. Carroll's contribution to the M.I.T. Conference. This paper has been considerably helped by his conceptualization.

dominate the formal authority system; multiperson groups (as well as individuals) take responsibility for problems.

But "amorphousness" of organizations need not be seen as simply a wastebasket category into which only unclassifiables are thrown. We can treat this class positively as a distinct form generated by frequent inputs of novel, complex, ill-structured tasks.

The kinds of problems that occur at these upper levels differ in two major features from those that happen at lower, more programmable levels. First, the *existence* of a problem is not well defined. That is, it is not always clear to which parts of the environment the decision makers should be attending. Identification of the problems is often more difficult than the problems themselves. We often characterize the quality of top management as much by how well they choose problems as by how well they solve them.

Secondly, the nature of a *solution* is not well defined either. It is often not at all clear when a problem at this level is "solved," or what the nature of the constraints should be.[7] In these areas, managers can find no algorithmic solutions. They must rely on their experience, intuition, insight — in a word, they must rely on heuristics.

If we follow our argument of long-run invariance between task and structure, we should expect a grossly similar structural form at the organizational level to which such ill-structured environments present themselves — notably the tops of organizations. At these upper reaches, the confrontation between classical hierarchical structures and committee structures seems to have been long since resolved in the com-

[7] See Minsky's definition of ill-defined problems, M. Minsky, "Steps Toward Artificial Intelligence," *Proc. I.R.E.*, 1961, *49*, 8–29. An extension and elaboration of this idea is offered by W. R. Reitman, "Heuristic Decision Procedures, Open Constraints, and the Structure of Ill-Defined Problems," in *Human Judgments and Optimality*, M. W. Shelly and G. L. Bryan (Eds.), New York: John Wiley & Sons, Inc., 1964, pp. 282–315.

mittee's favor. At least it is our observation that, in 1966, large American companies are run more by groups than by individuals. What then would be the nature of computer programs that could operate in these areas?

Continuing our mapping of organization structures into computer-program structures, we propose that the programs required here are those with heuristic and "participative" EP's. The heuristics will be necessary because there will be no algorithmic solutions to the decision problems at this level; the participative structure will be necessary because no single part of the EP will completely control any other part.

Programs of this nature already exist in a weak sense. They exist in exactly those areas where computer technology is beginning to have its greatest impact upon organizations — in real-time systems. There seems to be a fuzzy line between those systems that are real-time and those that are not. We consider "real-time" to be a relational concept. That is, a system is real-time or not with respect to the temporal constraints of its environment. For example, the missile-tracking programs of the NORAD system are real-time with respect to the intercept system. They operate quickly enough to allow the system to effect the process it is observing. Another facet of NORAD — the satellite tracking system — is not as easy to define. Inputs to the system are radar observations of the hundreds of satellites currently in orbit; output is a schedule of predicted sightings — a timetable for the radar stations telling them when and where to expect satellites for the next few days or weeks. The radar observations are "batch" processed; they are accumulated for many hours and then processed a few times each day. With respect to the high-speed missile-tracking programs, this is a *non* real-time system; with respect to the radar stations that need to know where to look tomorrow, it is real-time. What we are saying then is that every system is real-time with respect to something. To reconcile this odd conclusion with our intuitive notion of real-time systems, we offer the following operational definition. Let *t* be the amount of time that the computer could suspend computation — i.e., the "stop" button is pushed — and then resume without changing any of the inputs or outputs in the system. The shorter *t* is, the more real-time the system is. A one-second interruption of the control programs for a missile launching is intolerable. It would change the inputs to the program because the missile's path would change, and it would change the outputs from the program, because when the program resumed it would have to compute drastic corrections. On the other hand, a three-hour delay in the running of a payroll program would not change any of the inputs or outputs. We do not usually call a system "real-time" unless *t*, the tolerable delay time, is at most a few seconds, and more often a few milliseconds.[8]

The low tolerance for total interruption that we have used to define real-time systems imposes a requirement for the kinds of complex participative heuristic programs that we discussed earlier. The heuristics show up in priority decisions about how to handle a potentially enormous combination of input and output states, in procedures for detecting and correcting input errors and pathological conditions, in the strategies to minimize the probability that the system will ever grind to a halt. The participative features show up in the ability of any of the subparts of the EP to call upon any of the other parts, in the capacity of any subpart to interrupt any other, and in the availability of a common data base to all parts of the EP.

So far in our discussion we have distinguished object from information processing. But that distinction is incomplete. Additionally we need to distinguish the work-output processes of an organization from several types of "support" processes that take place. Some companies produce objects, while others "produce" information. Usually the two are combined,

[8] See the discussion of slow-time and slow, moderately fast and extremely fast real-times in A. Opler, "Requirements for Real-Time Languages, *Communications of the ACM,* 1966, 9, 196–199.

though often trivially. New automobiles carry information booklets. Insurance policies are printed on paper. Sometimes the mix is not trivial. Newspaper publishers significantly process, as outputs, both objects and information.

But behind these output processes, whether the output be object or information, one always finds some supportive informational activities carried out elsewhere in the structure than on the output line. Some of these informational subsystems are "directly" supportive of output, e.g., instructions to lathe operators, work schedules, etc.; some of them are more distant, e.g., payroll processing, capital financing.

In the human realm, some cases of direct informational support of output are excellent cases of real-time noncomputer information systems. For example, the lathe operator, within his skin envelope processes both object and information *in real-time*. But payroll information about him is batch collected, stored, and processed in another temporal world. It is only loosely bridged to the output processing that has occurred. The lathe operator's temporal world is bounded by machine speeds, production schedules, and the like. The payroll clerk's world is bounded by the tradition of weekly deadlines. The two meet twice — when the pay envelope is received by the lathe operator, and more importantly when the operator's time data are periodically batch fed to the office where they are accumulated in larger batches and finally processed on a schedule barely related to when the work was done.

Suppose, however, we could put payroll activities on "real time." In such a real-time information-processing system, we should have married object processing to some relevant information processing. The two activities would be interrelated far more closely than by the fragile bridge that had existed before. Though we are not touting such a system, one could presumably deposit funds to an employee's account immediately upon completion of *any* convenient unit of work.

In some sense, it is at the lowest levels of the organization structure that the organization's real-time work gets done. The bottom level of the manufacturing firm is the production line. The lowest ranked employees are characteristically closest to the object or information that is the organization's business. It is the blue-collar workers who bolt on the physical bumpers within the time limits imposed by the production process. As we go up hierarchy, we can not only define managerial rank by distance in levels from the organization's product, be that product an object or a form of information, but also by "distance" from real-time constraints.

Real-time human systems then existed before computers, but time wasn't always as real as it could be. There are structures that can work within existing time constraints more cheaply, more accurately, and more rapidly than people can, so that information can be processed in smaller batches or even continuously, and over a wider range than people can, so that more bits and pieces can be interrelated within existing time limits.

Real-time information systems can thus be treated as being equivalent to organization structures at almost any level of complexity. Real-time computer control over single machines has already been with us for some time. The recent interest in real-time "total" information systems can perhaps be thought of as an extension of real-time computer programs into much more complex multilayered, multidimensional structures. In combination with rapid communication devices, such systems can extend broadly through space as well as time, acting immediately on X and Y on the basis of information just received from Z.

Two current examples of real-time object/information processing

Consider two examples of moderately complex real-time systems currently in operation: A machine-shop control system, and the Westinghouse Telecomputer Center. The first operates at the object-processing level in the organization; the second at a much higher informational support level. In the machine-shop case, electronic circuit boards are being assembled. Upon its arrival at a drill station, a part sends a signal to the computer. The computer sends the appropriate instructions for that part to the drill. The drill drills and the part moves on. At some later time, the part arrives at an inspection station. Once again the computer is notified and sends the appropriate instructions for that particular part to the inspection station. If any of the holes are incorrectly drilled, the part is rejected and the information file on the drill that drilled the bad hole is updated to include this latest failure. If the failure rate for that drill is excessive, remedial action is taken. This same sort of thing happens for mounting of parts, etching of circuits, electrical and logical tests, etc.

This system uses multipurpose drills, inspection stations, etc., in a continuous one-at-a-time manner. Setups for batch processing do not exist. Parts are processed as they come along, in the order that they enter the system. This is a not too commonplace example of a real-time system operating at the basic production level. This kind of process control has existed in continuous manufacturing (i.e., oil refineries) for some time, but the technology for extending it into a wide range of noncontinuous manufacturing is just being developed. In many environments, it becomes quite costly to replace the EHM part of man.[9]

The Westinghouse Telecomputer Center operates on a much broader real-time band.[10] The applications of the TCC

[9] We agree with Simon's comments about the cost of flexibility in automation, H. A. Simon, *The Shape of Automation,* New York: Harper & Rowe, 1965, pp. 39–42.

[10] This description is taken from Robert D. Cheek, "The Westinghouse Telecomputer Center" (mimeographed), Westinghouse Electric Corporation, Pittsburgh, Pa., 1964. For further discussions of the TCC, see William Strauss, "The Westinghouse Telecomputer Center; a Preview for Managers," *Industrial Management Review,* 1965, *6,* No. 2, 65–69; and "How Computers Liven a Management's Ways," *Business Week,* June 25, 1966, 112–113ff.

are intimately involved with Westinghouse's nationwide teletype system. The Westinghouse teletype network is one of the largest industrial communication systems in the world and serves hundreds of separate terminals in the U.S. and Canada.

The Telecomputer Center handles the routing of all traffic on the network. A message originating anywhere in the system is transmitted to the computer, identified, placed in a message queue for the appropriate outgoing line, and eventually transmitted. The Center processes or redirects many thousands of messages each day.

Some of the incoming messages are directed to the computer itself. These messages activate the programs devoted to the automatic processing of teletyped orders and maintenance of inventory records for all standard or industrial products stocked in warehouses throughout the country.

Any of the over 100 sales offices served by the teletype system can originate an order message. The address code of the message directs it to the computer order-processing programs. These programs locate the required items at the warehouse nearest the customer and generate messages to the warehouse that prepares labels, bills of lading, and packing lists on the receivers at the warehouse. At the same time, price extensions and sales taxes are calculated, and the invoice is printed. Inventory records are updated in the process and examined for reorder points. If a reordering point is reached, the applicable formulas are brought into play to determine the proper replenishment quantity.

For a routine order, no hard copies of either the order or the shipping instructions are produced locally. Processing time averages less than three seconds from the receipt of a complete order to the beginning of transmission of shipping instructions to the selected warehouses.

The order-processing section of the TCC constitutes a nonhuman replacement for a large human substructure at Westinghouse. It is a real-time substitute for a human real-time system of several classes of human clerks, inventory controllers, and supervisors. As its possibilities are explored and exploited, it is likely that it will put higher organizational levels on "realer" time, by supplying much more current information to human managers, and/or by making many more adjustive and adaptive decisions immediately upon receiving appropriate information.

Another major facet of the TCC is the corporate accounting department. The same computer that processes orders in real-time performs accounting functions such as payroll, stock dividends, monthly reconciliations, etc., on a conventional batch basis. Some specialized responsibilities of the accounting department are being moved to real-time status, and these developments may eventually lead to a real-time management information system. This real-time system is superimposed upon a bottom layer of human object processors. The developmental efforts on the system are aimed at reaching into higher information-processing levels in the organization, rather than into the object-processing levels.

VI. REQUIREMENTS FOR TOMORROW'S TECHNOLOGY

But even these systems represent only a very modest degree of flexibility when compared to the top-level committee. When we start to talk seriously about computer programs behaving in a way that may be interchangeable with higher-level managers, we must talk about less modest issues; about programs that think, adapt, and learn. The closest things we have to these levels of intellectual performance are the heuristic problem-solving and learning programs that are themselves theories of human cognitive activity. These programs not only do things that humans must "think" about to do, but also they do them in the same way that humans do them. It often happens that these descriptive models are also the best performance programs.[11] The development of chess-

[11] For a brief discussion of this tendency for artificial intelligence

playing programs provides an interesting example of the development of a theory into a performer. It became evident quite early in the research that no computer and therefore no human player had the computational power to evaluate an algorithmic solution to a chess position. So some heuristics had to be incorporated into the programs. These heuristics were essentially postulates about the rules that humans used when they played the game. They turned out to be very powerful. What started out to be a descriptive model of human cognitive activity may soon be a better performer than the human. In a similar vein, the forerunners of future heuristic EP's may be the descriptive models of top-level organizational decision-making that are currently under development.[12]

If this is to become anything other than pure blue-sky talk, one serious limitation of the heuristic programs currently in circulation must be overcome: their single-mindedness.

It has been observed that "when a computer program is purposive it is too purposive."[13] Programs don't become bored; they do not mull over problem B while solving problem A; they do not notice what they are not supposed to notice. If humanoid programs are required to replace humans in these top management areas, then we need even more flexible concepts in heuristic programming, especially in the area of attention direction and resource allocation. It is not yet clear how human these humanoid programs need to be. We do not yet know what human characteristics are truly

---

programs to move toward psychological simulation, see H. A. Simon, and A. Newell, "Simulation of Human Processing of Information," *American Mathematical Monthly,* 1965, *72,* Part II, 111–118.

[12] We refer chiefly here to some of the models outlined in R. Cyert and J. March, *A Behavioral Theory of the Firm,* Englewood Cliffs, N.J.: Prentice-Hall, 1963, and to some of the current doctoral research at Carnegie Tech.

[13] Neisser, U. "The Imitation of Man by Machine," *Science,* 1963, *139,* 193–197.

irrelevant or detrimental to problem-solving in complex, rapidly changing environments. Perhaps some "neurotic" human behavior that is obviously irrelevant to direct problem-solving is not at all irrelevant to the search and attention directing aspects of problem identification and definition.

In the purely theoretical realm, some programs are now being developed that forget, get sidetracked, interrupted, etc.[14] Eventually, we may see these kinds of programs in organizational settings; but currently, even the single-minded heuristic program is just beginning to appear in such areas as production scheduling and planning warehouse location. Some types of managerial positions are more apt to benefit from such programs than others. If a job shop program considers only the "rational" requirements of the task but does not notice the "irrational" things that a good foreman attends to, then its usefulness will depend upon the task relevance of the foreman's "irrational" behavior. As we go up the managerial hierarchy, this flexibility of attention probably constitutes an essential requirement of a good manager. We shall have to capture it in our computer programs if we are to make programs and presidents mutually interchangeable.

In concluding, it should be noted that in discussing technological interrelationships of tasks, organization structures, and computer programs, we have not raised two real and important issues. One is the issue of economic feasibility of these changes. The other is the issue of their social desirability. Technical feasibility, we have argued, seems to have become an imminent reality. Economic feasibility will increase, but at a rate we cannot predict. Social desirability does not seem to us so much a new issue of man-machine relations as an old one of the relationship between man and man in the presence of a new technology.

[14] Walter Reitman, *Cognition and Thought,* New York: John Wiley & Sons, Inc., 1965, Chapter 8.

SUMMARY

This paper tries to relate three large variables to one another: tasks, organizational structures, and computer programs.

We propose a grossly invariant relationship between tasks and structures, over the long run. Definable, programmable tasks generate orderly, constrained, hierarchical human structures. Ill-defined, novel tasks generate flexible, open, nonhierarchical structures. Top levels of organizations usually work on ill-defined tasks, often converting them into a programmed state. Hence we would expect them to be relatively loose and nonhierarchical and similar across industries. Lower levels are usually given programmed or partially programmed tasks. Their structures are likely to be constrained and hierarchical and also task-specific. So they may be less interchangeable with other structures.

Computer programs, we assert, are nonhuman organizational structures. They are composed of executive programs and open or closed routines which perform functions and relate to one another in ways analogous to the relationships among levels in human hierarchies.

Most computer programs used in industry to date have served as substitutes for lower-level, highly ordered human structures (EDP programs) or as supplementary aids for moderately complex middle-level structures (OR programs). Not much industrial use has yet been made of more elaborate, many level, real-time systems-type programs, or nonhierarchical (at the top) humanoid heuristic programs, both of which appear appropriate for certain classes of more complex, ill-defined tasks.

With the evolutionary application of these programming developments, we submit that a new wave of interactions and interchanges between organizations and computer programs will be triggered.

DISCUSSION

EMERY In this paper there is a reference to invariance. If you mean by this that all complicated tasks are broken down into a hierarchy of subtasks, I don't think there's much argument there. But if you mean the linkages, the network, and so forth, then it seems to me that this is not the case; because when one breaks down a complicated hierarchy into subtasks, the primary objective in doing this is to make the subtasks manageable. And what is manageable is a function of your ability to manage. In the case of an information system, it's a function of your information-processing capability. If you have a high-capacity processor, larger tasks become manageable in some sense. In payroll, for example, if the programs are for a small computer, they simply aren't the same sort of programs they would be if they were written for a large system.

KLAHR I agree. And we would assert that most of those small payroll programs are pretty much alike. When the capacity does increase, there will be different programs, but again they will all look pretty much alike.

LEAVITT We really should have underlined very heavily the "other things equal" in our paper — other things such as technology.

EMERY It seems to me that to a large extent it's the system that's arbitrary, because we don't know how to break it down. For example, two organizations might have fairly different organization structures — one might have basically a functional organizational structure and the other a project organizational structure — and I think it's hard to assert that one is better.

LEAVITT I think you're quite right. In a very gross sense, we're arguing that given enough time and given the same state of technology and a few other things perhaps, men will make about the same organization structures to solve about the same tasks. We talk about the concept of centralized or decentralized organizations. This is really a kind of macro-assertion, in the grossest sense. If you get much more precise in your descriptive terms, so that you differentiate the subparts of the organization clearly, then I think you will find a large amount of variation. There have been some experiments in communications networks at Carnegie Tech recently in which the experimenters have essentially left networks open for various tasks to see what kinds of structures develop. And they came to this conclusion: "Our data demonstrate clearly that the A tasks (which are of a routine deductive nature) lead toward centralization." By this they mean that the organizational form set up by the subjects in the experiments is a class of things which they call centralized organizations — according to their definition of centralization. On the other hand, the B tasks, which are nonroutine and have an inferential component, do not lead toward centralization, although this is a fairly broad statement.

MYERS You say that all organizations look alike at the top. In addition, you say that the participative system of management has pushed out the hierarchical approach. Your generalization, looking way ahead, suggests that future top managements are all going to be characterized by the participative approach.

LEAVITT At the top. And not participative in any conscious way. I think if one takes a look at the tops of complex organizations, one would generally find that they operate less hierarchically and more "participatively" than do the lower portions. Certainly more than one would generally believe from reading textbooks on organization.

There's no indication at all of complete uniformity in organization. In fact, there's the implication of diversity. The larger organizations are really multiple organization structures that take different forms at different locations. And the primary determinant is the nature of the task which that portion of the organization — that subunit — is dealing with.

CLARKSON It has occurred to me that there's something that has been overlooked as a topic in the study of the behavior of computers and people together in an organization. And this is the university computation center. It strikes me as strange that no one has used this as a possible source for testing such propositions and/or examining behaviors. For here you have not only the "beast" behaving, but some people maintaining it, and other people using it — or trying to use it — and also a diverse structure connected with it all. An interesting question might be, "How much effort is spent by each individual in trying to subvert the computation center's priority system?"

MC KENNEY It seems to me that the physical scientists and the men who are trained algebraically have been the first ones to use the computer to amplify their own scholarship. People who think in a less well-defined fashion, and in English perhaps, have had a difficult time getting into the computational center and have done so only through programmers or other well-trained people.

KLAHR It seems that one of the real problems in considering systems is not what to put in but what to take out. And yet it seems that nothing ever gets taken out of a language or a system — or at least very rarely. The justification for keeping things in is that people are still using them. But this is kind of a self-fulfilling prophecy — they're using them because that's all there is.

RESCHER One of the weak points of your paper seems to me to be in connection with the nature of structured tasks. You argue from the premise that where the problem task is structurally defined, one would expect to find some sort of sameness in the solution-producing structure. And then you go from this to what seems to me to be a quite different sort of thing; that is, to where you have no well-defined problem task at all. There the task is amorphous. And yet you still expect to find some sort of similarity in the solution-producing structure. I think that this does not necessarily follow. The examples you cite in support of that are weak, I think, because actually all of them deal with fairly well-structured kinds of things and the analogy to the amorphousness of certain types of industrial structures is not very good.

LEAVITT There is the question of whether you think of an amorphous task as a positive class of tasks or as the absence of tasks. What we described as an amorphous organization is a kind of specialized loose form. I submit that there are sets of tasks which are distinguishable by their amorphousness, by the fact that their boundaries are not clear. Perhaps this is a tautology, like, "All nonlinear systems are alike in that they're nonlinear"; but I don't think so.

FORRESTER It seems to me that there's a continuous gradation from the highly structured, precisely definable task into those called amorphous. The amorphous category is separate. It is only that we haven't yet been able, or haven't taken the time, to think about the criteria for the particular decision. This boundary between the structured and the unstructured is moving very rapidly. These things that would be cited as unstructured — capital investment policy, for example — are perhaps even now within reach of being structured. It seems to me it might be a mistake to talk about them as if they were a peculiar and separate class that can't be overtaken by this moving boundary.

LEAVITT I quite agree. This is really a dynamic process in which what is unstructured today is a structured task tomorrow.

FORRESTER The unstructured task is the one we don't yet understand.

LEAVITT Yes. We don't yet understand it to the extent that we don't have the technology to do the analysis.

FORRESTER The unstructured task is the one for which we don't yet understand the structure, but we have acquired enough of an intuitive feeling that we can deal with it. Beyond the unstructured tasks are those things that we don't deal with at all because we can't even handle them intuitively. Both of these boundaries are moving together, so that the tasks we aren't dealing with at all today, fall within the unstructured group of tomorrow; just as today's unstructured ones become the structured ones tomorrow.

DECARLO There is a sort of danger, though, in this notion that if unstructured tasks wait long enough, we'll be able to define them and then put them into programs and onto the machine. And the danger is that within corporations you have an almost implicit acceptance of the notion of the finiteness of the work to be done. Most managers don't think corporate existence is open-ended at all. So there is this fear that as you structure, you kind of run out of work. I happen to think that it's incorrect, but I think it's there.

To illustrate this better, I was thinking of our discussion about the university computing centers. There's an important difference between working in a university and working in a structured operation like a corporation. Project MAC, or any of the university functions, is planned to contribute a service to a group of people who are themselves highly individual; and it's to preserve their individuality that they constitute a university. So the computer exists for the service of an individual. Now, let me go from Project MAC to IBM's Poughkeepsie operation and consider, say, one of our best design development engineers. How does he look at the computer? I think he has a completely different view. He sees it in two dimensions. First, in his professional work he sees the computer taking some of his guesses and going through a design automation process and feeding them back until gradually a stable local solution is obtained. So he sees the machine as something which, depending on whether he's a pessimist or an optimist, is either taking his job over or expanding it. The other way he sees the computer — and the one which I think this conference will in the long run be concerned with — is as the implementer of a system of control. The computer just happens to be the mechanism that allows the organization to keep him under control.

I would conclude that in the university environment you try to encourage creative activity and create the illusion that each user has a complete machine. On the other hand, in an engineering laboratory in business, you try to achieve a collective purpose within which individuality can operate. In this context, the man sees the computer as both a professional "extender" and as the manifestation of a control device. The danger in the first case is that in the long run the freedom to explore wider technical choices may very well be denied. In fact, I think that by and large in the computer industry, it is probably going to be denied by over-rigid system designs within the design process itself.

CAESAR I share this view completely. And yet I'm still bothered by the analogy in the paper, especially at higher levels where participation does not make sense to me in terms of how management really operates in a corporation. You may have discussion back and forth, for example, but the need to make a decision and where that decision rests is not as unambiguous as we like to think it is.

LEAVITT You're not going along with the notion that presently unstructured decisions are dealt with by a kind of relatively vague set of priorities — like discussion, consideration, and reconsideration? And decisions kind of emerge rather than get made by single individuals?

CAESAR I take some exceptions to that. I think you have that process but you don't usually vote on the decision. The final responsibility is pretty clear, not ambiguous or uncertain. But you must make the decision within a certain time-frame. You can label the process of how "he" (the top manager) comes to that conclusion as being relatively unstructured. But I don't really think it is in the last analysis, because "he" must make it.

LEAVITT The decision is still made, then, by one man in a responsible role?

CAESAR I see very few committees operating. You know — they get together and talk, but I don't see that the decision comes from the committee at all.

LEAVITT So the decision is made by an individual? I guess I don't agree.

GUEST In discussing unstructured problems at the top management level, I have been surprised by the amount of participation that I've seen. In the early days of industry, there were probably a lot more one-man decisions; but now these problems are dealt with by committees. A good example of the amorphous kind of problems faced by top management is the governmental and societal pressure placed on automobile corporations for safer cars. This has been something to which the automobile industry apparently has not paid too much attention until recently. I suspect that now, at the very high levels of General Motors, there is probably a lot of participative discussion; and yet the problem itself is a very amorphous one because it involves all kinds of inputs—public pressure, legislative action, cost of safety devices, etc. This is a concrete example of the amorphous sort of decisions that have to be made at the top.

WILLIAMS Who deals with the amorphous problem? I think a lot of organizations use consultants for this and allow them to work on dimensions that haven't been explored before. If you know what the problem is and you tell the consultant—then he isn't in a true consulting role. He is just an implementer of a specific program. But if the organization is ready to talk about ill-defined issues, then the consultant's role is a real one. I think you could study a true consultant and get considerable help for the program that you're talking about.

LEAVITT The top management staff groups now have many of the same kind of functions.

KLAHR In consulting firms that use computers heavily, you are, in effect, buying a combination of a human and a computer decision. If a company lacks the resources and the decision-making capacity, it can go to a small consultant who uses computers and obtain very well-defined decisions. Maybe the very complex models of decision-making in the future will come from the consultant firms.

DECARLO In an organization, what is called participative management is really an attempt at a particular cell level of the structure for a man to listen to many different points of view, not only to get a consensus, but to spread the risk of the decision-making. After all, as a professional manager, he has such things as risk-buyers and there is no reason why he should make a decision. As a matter of fact, with all this seeming participation going on, I suspect that more often than not time moves by in such a fashion that the necessity for making a decision disappears, because the events transpire in such a way that a decision is no longer necessary. I think the artful manager is the one who knows when he has the time to wait and let the decision be made by happening and when he has to decide to take action today.

DELEHANTY It seems to me that Dr. DeCarlo's view is an optimistic one for those who like structured problems. As he points out, there is some risk. The risk to the individual is very great; the risk to the organization may not be as great because of the bigger denominator.

Another thing is that the search for a solution to a problem is costly to an individual because this solution will then bear his label. If it fails, then he has incurred a risk. This suggests to me that there ought to be an active striving for programs which will search among alternatives impersonally and then present a "menu" that can either be validated or not. There's a real advantage in having such a search — or, if you want, a decision process — made less personal.

DeCARLO Of course, it seems to me, the most ardent people for computer techniques —aside from the programmers and professionalists who enjoy playing with them for their own sake — are the top managers who would just love to have such a "menu" to blame.

LEAVITT I think of the members of the executive committee of one company that I visited, who said, "We don't have to make decisions quickly. We don't make any decisions that take less than six months to make; the problems we deal with are open-ended." They didn't see any computer programs that might be helpful to them in their decision process.

EMERY Because the response time of the decisions they make is long, they feel that knowledge about what's going on in the world right now is unimportant and therefore they don't need a computer. Well, I would agree that they may have a long response time, and that very timely information may not be important. But the thing that makes the computer very valuable, I would assert, is that it allows you to look at more alternatives.

KLAHR As a word of summary, I'm still a little disturbed by some of the associations that have been suggested. In our paper we in no way implied that the organizations that we described were somehow model organizations — that there were no such things as politics or coalitions or multiple motivations. All we were doing was trying to describe the kinds of processes that take place, or that seem to take place, when multiperson groups try to tackle complex, unstructured problems.

We also suggested that although one tends to find these complex, unstructured problems mostly at the top of organizations, this isn't the only place in organizations that one finds them; one presumably could find such complex, amorphous, unstructured problems in research operations or in the activities of many staff departments. Also this kind of unstructured decision-making does not necessarily pervade the entire organization. In fact, in large complex organizations some parts are very highly programmed and the decision is very easy to specify, while in others, quite the reverse is true.