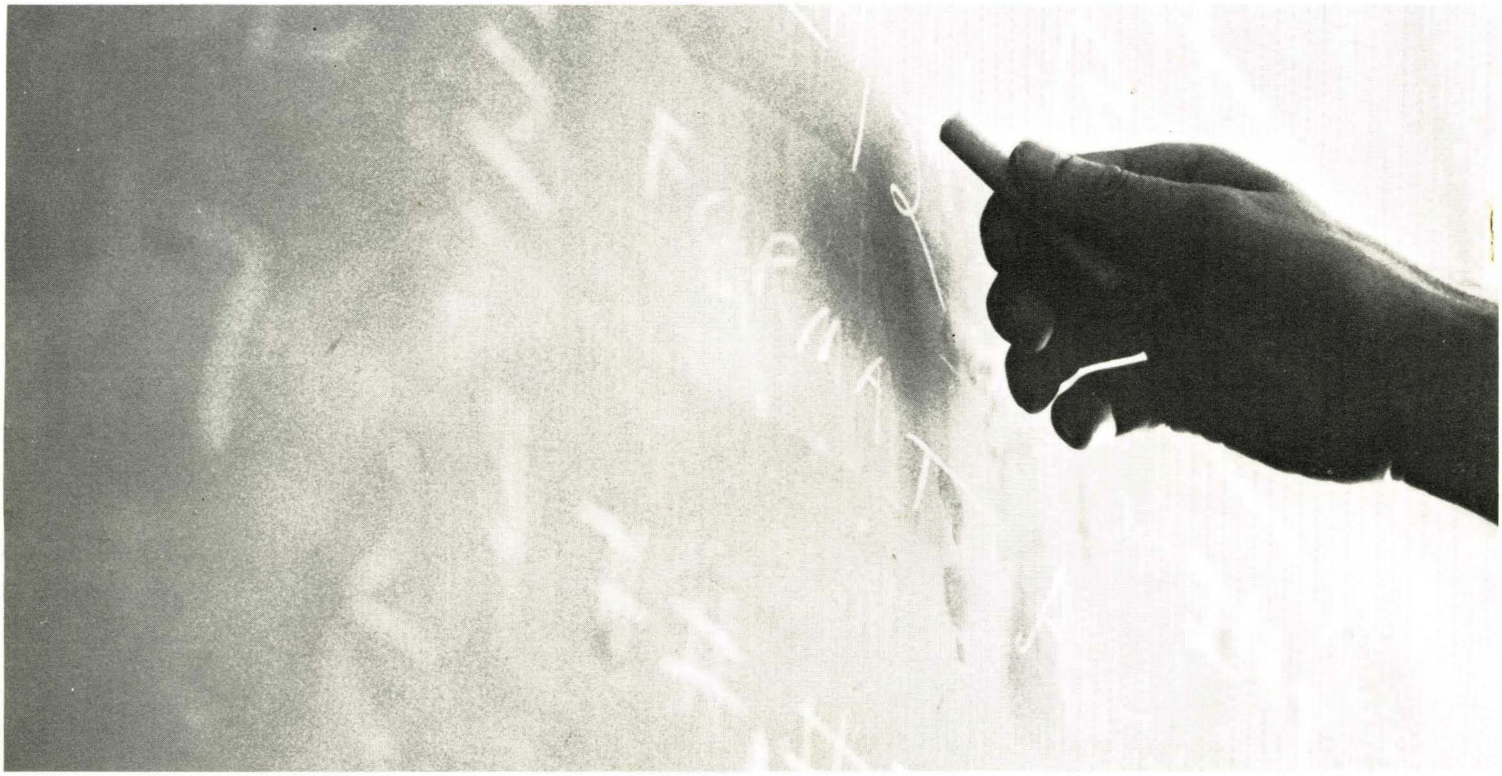
SDC

SP-2169/000/01



Designing a Machine Partner--
Prospects and Problems

Aiko Hormann

15 October 1965

SP-2169/000/01

SP a professional paper

Designing a Machine Partner--Prospects and Problems*

by

Aiko Hormann

15 October 1965

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA

*Research on the program system described in this paper is supported in part by the Office of Naval Research, Contract No. Nonr-4745(00).

REV. 4/64

ABSTRACT

This paper discusses some possibilities for extending man's intellectual and creative power through "partnership" with machines of increased responsiveness and sophistication. Some of the general requirements for a machine partner are stated--requirements that are likely to differ depending on the human users, their purposes, and the situations. Some of the different problem domains are discussed, and reasons are given for handling some problems by machine alone, some by a man/machine team, and still others by man alone.

Depending on the purposes of a man/machine system, and on the techniques used in building it, machine behavior can range from the accurate and efficient performance of very specific tasks to the exhibition of some aspects of "adaptive" behavior and other higher-level intellectual capabilities. The paper also discusses the conceptual and technical difficulties that must be overcome before the machine can become a partner to man, rather than his simple-minded servant.

There are a number of psychological theories and computer techniques that may be applied in developing a machine partner. One is the "cognitive map"--an internal model of an external world. A machine can be programmed to construct, use, and modify its own cognitive map as it gains experience, and thus to increase its effectiveness in dealing with its environment. Another possible technique would be to realize machine functions analogous to those believed to be operative at the human subconscious level. Three commonly observed phenomena are discussed, as well as ways in which computers might be made to exhibit similar behavior. The purpose of doing so would not be to explain such phenomena, or to imitate human behavior, but to find fruitful methods and techniques for increasing the machine's problem-solving capabilities. One new idea introduced in this connection is a special way in which a conventional serial-memory computer might be connected to, and interact with an associative memory (content-addressable memory). The technical feasibility of this idea is confirmed, and ways in which its advantages could be used in complex problem solving are discussed.

The paper then suggests the possibility of having "ideation sessions" with a machine partner in problem situations requiring new ideas. The machine's contribution might be greatest in the second phase of problem solving, i.e., the judicious evaluation of ideas and the selection of fruitful ones from a large volume of diverse ideas, both good and bad.

Last, the paper discusses some hurdles that must be crossed before man can enjoy the benefits of machines worthy of the name "partner."

DESIGNING A MACHINE PARTNER--PROSPECTS AND PROBLEMS*

Aiko Hormann
System Development Corporation
Santa Monica, California

"Man is always more than he knows or can know about himself. . ."
--- Karl Jaspers

Man often makes decisions based on something he does not clearly understand and cannot define, and that we call--for lack of a better term--intuition. Personal biases and irrational actions have often been blamed on this ineffable quality, but just as often it is the source of human ingenuity and imagination and creativeness. There is no substitute for these creative qualities, but we have many evidences that man's functions can be greatly extended by computer techniques. This extension in the intellectual domain may be somewhat analogous to the extension of man's physical power by the machines and techniques of the Industrial Revolution.

In the frontier work of science and technology, man/machine partnerships can help to supplement and nuture the creative qualities in man by spurring him to explore his hunches, helping him to see his problems from different angles, weighing different alternatives in terms of their consequences, and minimizing his personal biases. So far, however, most man/machine "partnerships" have been based on a division of intellectual labor in which the routine work was done by the machine and all higher-level thinking by the man.

*Research on the program system described in this paper is supported in part by the Office of Naval Research, Contract No. Nonr-4745(00).

This trend is now changing, as Dr. Licklider recently pointed out in "Man-Computer Partnership" (Licklider, 1965). While the conventional uses of computers will continue to be valuable, closer interactions between men and highly sophisticated machines can accelerate advances in science and technology and open up new areas of exploration.

Attempts to develop man/machine systems of increasing sophistication (and the advanced machines they require) have been under way for some time and have resulted in useful and impressive advancements, but there are still many obstacles in the way. In the following, I shall try to point out some future possibilities--some different ways of utilizing such partnerships--on the basis of what is being done today. I will take an "inside look" (in terms of structured computer programs) at our future machine partner and discuss how man's creative activities may be affected.

What do we want from our machine partners?

In a sense, we humans are "playing God"; within our technological and conceptual limitations, we can mold (by programming) the style of the "machine mind." It is essentially up to us (the users and designers) to specify those qualifications that are desired for our machine partners. Do we want many human-like characteristics? Do we want an obedient but inflexible servant to take over only routine work with superhuman speed and accuracy? Or do we want an intelligent adaptive assistant or partner, which can adjust to our needs and problems and utilize its past experience? Since much of man's

intelligence is attributed to his capacity to learn and to use his past experience efficiently, it seems reasonable to assume that a machine partner worthy of the name should also be endowed with some ability to learn. I believe that a team consisting of a man and an intelligent, adaptive machine can be far more powerful than either of them alone, in many areas of scientific/technical endeavor. Many complex and difficult problems in such areas are a composite of the well-defined (i.e., those for which rules and goals are precisely known) and the ill-defined, and of some easy and some difficult subproblems. For such many-faceted problems, the human problem-solving process usually takes several courses. The problem-solver will follow rules learned in formal education and rules of thumb gained by experience; he will form hypotheses and modify them; he will use imagination and hunches.

Forming and testing hypotheses and exploring hunches seem, on the surface, to be activities that can only be accomplished by man, but they contain subprocesses that can be specified in computer programs even at the present level of our technology. As time goes on, larger and larger portions of these processes can come within the province of the computer. An optimal combination of man/machine talents, therefore, can be expected to encourage the human user to concentrate on the parts of the problem-solving activity of which he has no formal description. A singular advantage accrues in the presence of such a system, for with it, a creative man can maintain the momentum of his thinking; less frequently will the continuity and impetus of his thoughts be interrupted by tedious hours of noncreative work. Additionally, the human user can conduct

"ideating" sessions with a machine of this kind. It may turn out that man will prefer "ideating" with a machine to "ideating" with his human colleagues; he can express all sorts of ideas, even wildly impractical ones, without fear of ridicule, and judgment and evaluation can be deliberately postponed until enough evidence, pro and con, has been gathered. (See "Have an Ideating Session With Your Machine Partner," in the last portion of this article.)

Before proceeding, I should warn the reader that I have been using the general term "machine" when I really have been talking about a <u>programmed</u> computer. I prefer the general term, since a general purpose computer, when programmed, becomes a special purpose computer, which in turn can be realized (in principle) as a physical entity, a "machine." (E.g., a robot, with its physical parts and movements, can be simulated by a computer.) Also, I have been and will be using some anthropomorphic terms such as "learning," "adaptability," and "experience," but they should be understood to have restricted meaning when applied to machine functions. They are used for convenience and economy of words in conveying, <u>intuitively</u> and <u>informally</u>, ideas and concepts about machine behavior and capabilities.

Much of what follows will describe the machine partner in terms of my own concepts and assumptions on the subject of intelligence and learning. These concepts have been formed gradually over many years through reading books and articles, through discussing the subject with many people, and through my own interpretations and introspection. Therefore, it is impossible to give exact sources of information. However, most of the assumptions used here can be found in one compact book, <u>The Psychology of Thinking</u> (Thomson, 1959).

Given this mutual understanding about word usage and assumptions, I think we can group the basic requirements for an adaptive machine partner into three major categories.

1) Adaptability within a fixed problem context. For the solution of extremely difficult and complex problems--either well-defined (like chess) or ill-defined (like socio-economic problems)--man can prepare the machine initially, utilizing to the fullest the existing programming techniques and as much knowledge as he has of the given problem. If the machine is equipped with efficient learning capabilities, it can then "learn" the rest of the methods and techniques required for arriving at the solution through interaction with the men. Clearly, the man will be learning too; initially his incomplete knowledge of the problem may restrict the area and method of exploration, but as he gains more information and interacts with the machine--examining and evaluating the consequences of his assumptions, exploring his hunches, and analyzing the data--he may get a clearer picture of the problem situation, may be prompted to modify his problem formulation and his methods of attack, and repeat the interactive process.

2) Adaptability to the needs of many different human partners and to many different situations. Different problem situations often require different terms for description, and different methods and processes. Also, each person tends to use the same terms and processes in somewhat different ways. A machine can be made to adapt to individual specialties so that each user will think of the machine as "his" partner, even though it may be working simultaneously with many persons.

As the work proceeds, the team of scientist and machine will build up a set of "in-group" terms and expressions, as well as a repertoire of skills. In time, the style and degree of communication between them will change radically. This phenomenon can be observed in two people who have worked together on a project for some time. They begin to assume a great deal of common knowledge in their communication, and stop going back to basic definitions when they discuss their problems. An outsider to the project, hearing such conversations, may be baffled.

3) Adaptability to unknown or unpredictable environments, and in environments inaccessible to man. After millions of years of evolution, man is well adapted to the environment of Earth. But a sharp deviation from an earth-like environment will make man quite helpless. Other planets, deep sea areas, and some artificially created conditions on this earth (e.g., high radioactivity, near-absolute-zero temperatures, etc.) although not directly accessible to man, can still be studied through partnership with an adaptive machine. The machine partner can be in the environment but man need not be. Distance between them need not hinder close interaction. The machine will be able to make many decisions on the basis of its interactions with the new environment; it will detect regularities in physical objects, or in recurring patterns of events, and thus help the man to form hypotheses about the new environment. Man can then learn to cope with that environment, or to alter or control it.

Within each of the above categories, however, different situations and purposes call for different sets of characteristics and skills in the machine partner. We must study the various problem situations and decide what our purposes are.

The nature of problem domains and corresponding involvement of man, machine, and man/machine.

When we recognize a problem to be solved, the main question is, "What do we know about the variables that enter in describing the problem situation?" This question must be answered insofar as possible before we can consider problem-solving methods. First, are all the problem variables known and distinguishable? Only some? Does it seem likely that new variables will show their effects as the work progresses? (For example, in many socio-economic problems, factors not thought relevant initially may turn out to have a subtle influence on the whole or on a part of the problem.) Secondly, we must ask questions about the behavior of known and distinguishable variables. Is the behavior of variables completely known? Can it be expressed in a functional equation or plotted as a graph? Is the behavior predictable in a probabilistic sense? Is it predictable only within some boundary limits? Or is it completely unpredictable?

Figure 1 represents my subjective (and necessarily over-simplified) view of the nature of problem domains and corresponding involvement of man, machine, and man-machine. The three shaded areas represent those problems or tasks that can be attacked by man alone (dark grey), by man and machine

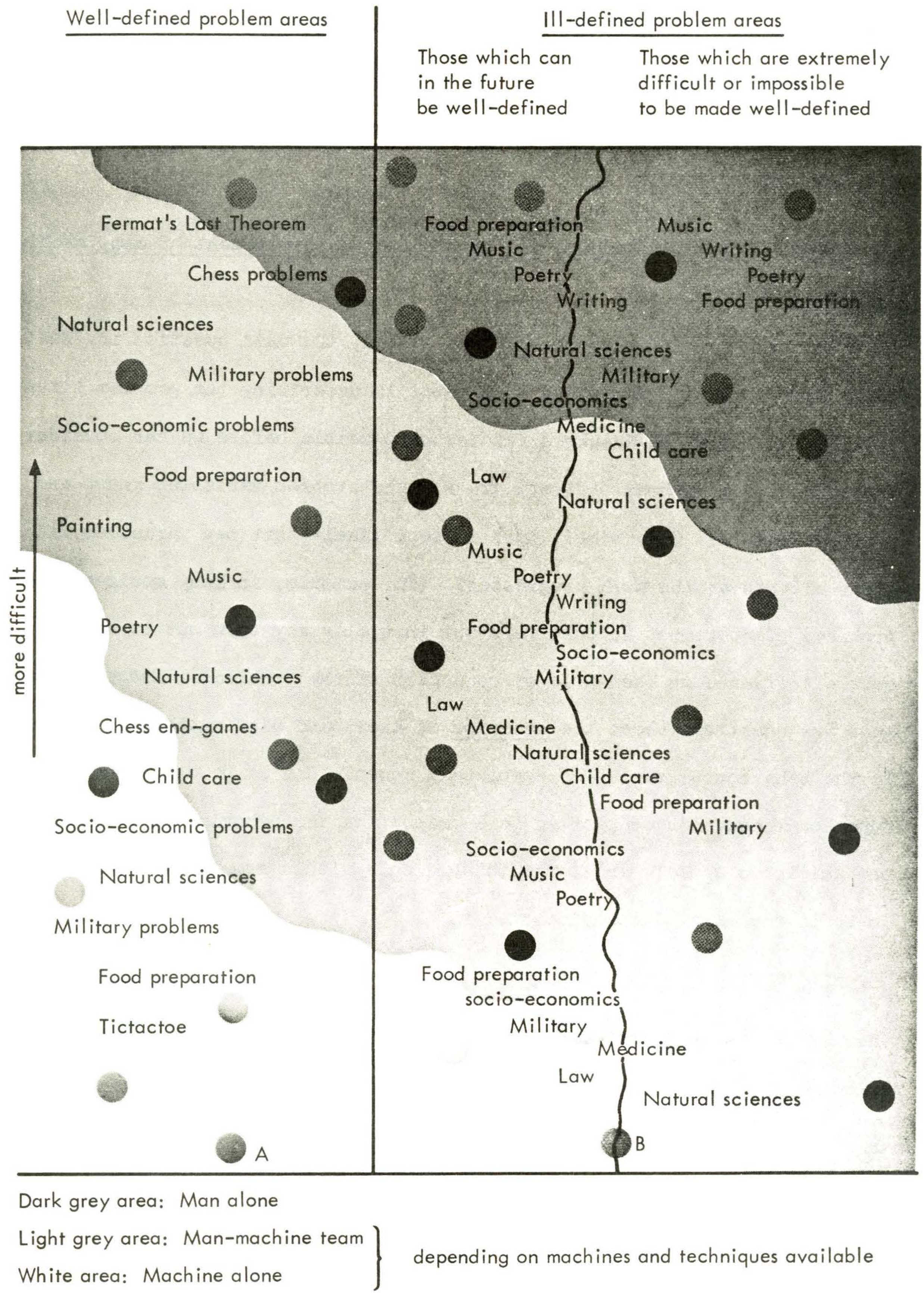


Figure 1. The nature of problem domains and involvement of man, machine, and man-machine

together (light grey), and by the machine alone (white). The wavy lines indicate that the distinction is not really clear or final; the lines are likely to be shifted as we increase our knowledge and as more and more sophisticated machines are developed.

1) Man alone (dark grey area):

   Some problems may remain man's alone because they require a high degree of intuition, imagination, personal taste, or the "human touch."

2) Man-machine (light grey area):

   Some problems can be handled by a man/machine team, either ideally or awkwardly, depending on the problem and the sophistication level of the team.

3) Machine alone (white area):

   Some problems or tasks can be handled by a machine alone. Performance by a machine alone may be either desirable or imperative because the task must be done in a hostile environment--extreme heat, radioactivity, etc.--or because the task is routine and monotonous, or because it requires a degree of precision or speed impossible to expect of humans.

The non-uniform nature of these three areas (indicated by the sprinkling of all three shades throughout) and other peculiarities are explained below.

The left column represents well-defined problem areas and the right column ill-defined areas. The latter is further divided (by a wavy line, indicating an unclear distinction) into problem areas that can be well-defined in the future, as we come to understand their nature better, and problem areas that are either extremely difficult or are intrinsically impossible to define. Well-defined problems are not necessarily easier to solve than ill-defined ones. For example, Fermat's Last Theorem has not been proved or disproved for many generations, and the game of chess has no solution in that no known sequence of moves will guarantee a win (or even a draw) every time.

Some problem categories, such as military and socio-economic problems and some creative/artistic fields, are listed in both left and right columns, and some are repeated in both the upper portion (more difficult to solve) and the lower portion (less difficult) The categories are repeated because some parts of some of these problems, or some phases of the task-performing processes, are or can be made well-defined, even though the over-all problems or tasks are not well-defined; similarly, some parts or phases of the same problem can be extremely difficult, while others are relatively easy. The same reasoning explains the presence of patches of other shades in each area.

Some problems are difficult not because of their complexity, but because of the perfection or precision desired (e.g., a slight error can be a matter of life and death). On the other hand, some tasks allow a wide

range of satisfactory performance. This range of acceptability, either wide or narrow, may reflect the "relative" or "personal" nature of some problems.

The relative and personal nature of some ill-defined problems (i.e., those for which rules and/or goals are not precisely known) are among the reasons such problems are so difficult to analyze and evaluate. In fact, imprecisely defined rules and goals for a particular line of endeavor (e.g., painting) are among the sources that give rise to artistic/creative expression. The artist can create new rules and goals, or old ones can be given a new interpretation, even changing the criteria of aesthetic appeal.

To explain the two dark grey circles marked A and B at the bottom of Figure 1, let me relate a practice I observed a few years ago in a department store in Tokyo. On each floor, standing at each escalator's starting point, there were two pretty girls, one on each side. These girls smiled and greeted each customer as he passed, and wiped the escalator rails. Such a task is extremely simple to perform and could have been done by a mechanical hand if sanitation were the main purpose, but performing it mechanically could not produce the atmosphere of "personal touch and care" to each customer created by the smiles of those girls. Here the "human element" is the essential ingredient.

Notice that one dark grey circle is in the "well-defined" section and the other in the "ill-defined" section. The task of wiping an escalator rail and smiling <u>can</u> be defined precisely, giving exact movements of the muscles

of hand and face. However, the same task can also be defined simply but imprecisely as "to please a customer at the escalator." What do we know about pleasing different individuals? (To be sure, it is a very personal notion; a wife might be irritated if her husband smiled back at those girls.)

Methods and approaches used in making a computer do things or behave differently.

Table I allows a quick comparison of some different ways (1 through 4) of specifying a machine's behavior, with a corresponding characterization (a through g) of each. Columns 1 through 4 are not to be taken as independent or mutually exclusive; methods and techniques applied in the right-hand columns also make use of methods and techniques applied in the left-hand ones. In a loose sense, these columns, from left to right, suggest the historical development of computing.

1) Completely predetermined specification. All the steps in the processes required are predeterminable. That is, in problem solving, all the steps in the solution procedure must be known in advance even though the answer to a specific question may be unknown. (As an example, take the set of programs for a payroll calculation. All the steps necessary for the calculation are known, but an answer to a specific question, "How much will Mr. X be paid this week?" can be answered only after the execution of the program.) This is further divided into two categories:

Table I. Different ways (1a through 4) of specifying machine's behavior with the corresponding characterization (a through g) of each.

	(1a) complete specification (one-to-one)	(1b) grouped or parameterized (one-to-many)	(2) feedback utilization	(3) past experience utilization	(4) secondary learning utilization
Problem examples	automated bakery shop	payroll calculation	traffic light control	learning machines (artificial intelligence research)	
(a) Kinds and amount of outside control after the preprogramming	Start the operation	Set the parameter values at the start	Feedback from the environment during the run	Recorded "experience"	In a form of suggestions and statements
(b) Nature of preprogrammed specification	Most problem-specific	Some generality	General rules for utilizing feedback	Can be more general	Can be highly general
(c) Degree of conditionality in the internal specification	No conditionality	Use of parameters increases conditionality	Added conditionality from the environment		
(d) Kinds and amount of self-correction or self-modication	None	None	Parameter values	Through information and program changes	
(e) Likelihood of observed behavior changing with time	None	None	Some changes	More changes →	
(f) Memory requirements in addition to preprogrammed portion	None	None	Not unless parameters increase	Record of past experience	Record of what has been "taught"
(g) Complexity of the system structure in both factual collection and process collection	Can be highly structured, but no dynamical changes		Tend to be more highly structured and allow more dynamical changes →		

a) One-to-one. If no variations are expected or allowed, one program handles one specific problem only. Since the program is custom tailored, it can be made extremely efficient, but a hundred problems require a hundred distinct programs.

b) One-to-many. Groups of operations or instructions that are repeated with minor variations are put together into subroutines or subprograms; slight variations are taken care of by introducing parameters that can take on two or more values according to the different cases of the problem. All the parameter values must be specified before or at the start of the execution. However, such values influence the machine behavior during the execution through conditional branches. (A single conditional branch in a program can make the machine behave, functionally, like two different machines in different circumstances.) This technique of parameterization combined with the use of conditional branches makes it possible for one program to take care of many problems of similar type (one payroll calculation program can be used for thousands of employees).

2) Feedback utilization. Suppose that the task environment includes some unpredictable elements in it. Then the steps necessary for the performance of the task are not all predeterminable. The decision as to what to do in this case is made on the spot during the interaction with the environment. (That is, feedback from the environment plays an important role in deciding the machine's behavior.) Automatic control of traffic lights, whose duration is

not predetermined but depends on the flow of the traffic, will have to use this kind of feedback information.

The nature of the environment and desired responses may require a special peripheral attachment and/or preprocessor--an input device to accept and pre-select the feedback information (e.g., a sonar or visual device, an analogue-digital converter, and/or some filtering or pattern-recognition device). (In a simulation laboratory, a human operator usually interprets the natural environment and inputs already processed data as feedback.)

3) Utilization of past experience (higher-order feedback). In (2) above, feedback allows the machine to "interact" with the environment dynamically, but the machine is not given a means to record and use its past experience. Its self-corrective behavior is only local and specific to the task; if the task were to be repeated, a machine of type (2) would repeat the same trial-and-error behavior before the proper adjustment was made (unless the feedback unit itself were made to set the parameter values and retain the same values).

With extra memory space for recording past experience, and special programs to handle its utilization, the machine can "learn" to avoid previous mistakes and will tend toward straightforward use of successful actions, without going through the groping or trial-and-error actions that were necessary at first.

Some behavior changes can be made at the level of parameter-value setting by a control program, but other changes require a higher-level program to change parts of the subprograms. This is conveniently done in stored-program

general purpose computers in which instructions in the program can be manipulated and altered by other instructions as if they were pieces of data. Internal control units can be hierarchically structured, and modification can be effected at various levels; we can write a program that will modify its own rules and instructions, or will modify the <u>way</u> it modifies rules, and so on. Thus, multilevel higher-order feedback units can be constructed so that they interact among themselves as well as with the environment. Some learning machines developed under artificial intelligence research show such characteristics. Later, I shall describe an adaptive machine I have designed, not because it is the best one, but because it is the one I know best.

4) <u>Secondary learning (learning from the experience of others)</u>. We learn about many facts and rules from other people and from books, without actually experiencing those facts and rules first-hand. Similarly, a machine can be made to accept suggestions about special methods, or suggestions about steps in a particular problem-solving context, or descriptions of a new task environment; it could be given a description of some processes of generalization in a form similar to a "lecture"--a series of statements interspersed with demonstrations and relevant questions. Clearly, these statements and suggestions given by a human tutor should not merely be stored. "Meanings" of such inputs must be analyzed and their implications studied; these in turn must be assimilated into the machine's memory in an organized way by establishing a useful relation to previously recorded information. Just how such "self-organization" should be achieved is a big research problem in itself;

one aspect of it is treated briefly in the section entitled, "Construction, Utilization, and Modification of a Cognitive Map," in this paper.

Primary and secondary learning are two modes, rather than kinds, of learning and in humans they are so intimately related and their effects so interdependent that they are often indistinguishable. However, it is convenient to use the separate terms in describing machine learning because endowing the machine with the capability to use secondary learning can cause an effect that is in significant contrast to learning through first-hand experience only. Intellectual development of the machine by a mixture of primary and secondary learning can be expected to be faster and more susceptible to human interaction than by primary learning alone. As an analogy, consider a student of chess; he might acquire a knowledge of the game from actual play (primary learning) only, or from play combined with reading and the tutelage of good players (primary and secondary learning). The second method would surely shorten the time it took him to gain proficiency in the game.

However, secondary learning depends largely on the ability to communicate. A machine capable of effective secondary learning must also be capable of sophisticated communication, including an easy-to-use language for humans and the corresponding conversion technique to the machine's internal representation, a two- or three-dimensional visual input/output, some form of sound input/output, etc. (There are excellent discussions by Dr. Licklider and Dr. Sutherland on the subject of man-machine communication; both talks were presented at the IFIP Congress 65.)

The foregoing discussion about specifying the machine's behavior, along with Figure 1 and Table I, can be used as a general guide to determine an optimal match of machine characteristics for a given problem situation. A mismatch of problem and machine sophistication is bound to cause some ill-effects; at a minimum, it will lead to costly operation--at worst, it will lead to chaos.

An example of an adaptive machine, programmed to use its past experience.

Here at SDC, we are working on a learning system (of computer programs and auxiliary devices) that incorporates the aforementioned features and techniques. The system is called Gaku, which is a Japanese word denoting learning. A simplified description of Gaku is that it is composed of hierarchically interacting feedback-loop units, each of which is equipped with general rules for decision-making and information-handling within its prescribed domain of authority and activities.

Gaku has four mechanisms, which are embedded within a master feedback unit called the mechanism coordinator (see Figure 2). This mechanism coordinator, as well as each mechanism, is designed to apply a basic cycling process that is useful in many situations where a problem-solver is forced to make a decision or take an action with insufficient information. As the situation unfolds, previously inadequate actions may be corrected or adjusted by utilizing additional information that becomes available as a consequence of the previous action.

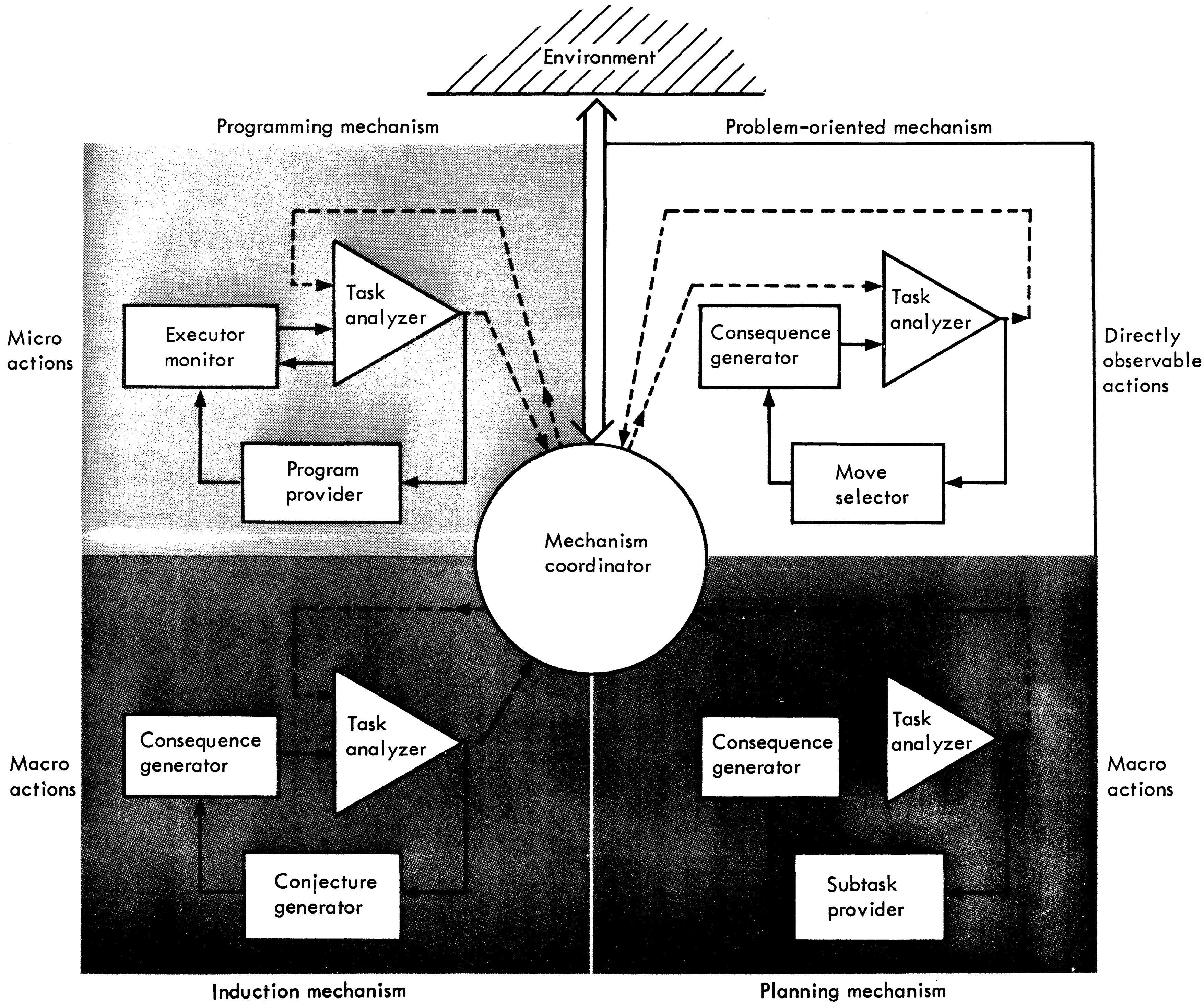


Figure 4. Closer look at the four mechanisms showing the basic cycles.

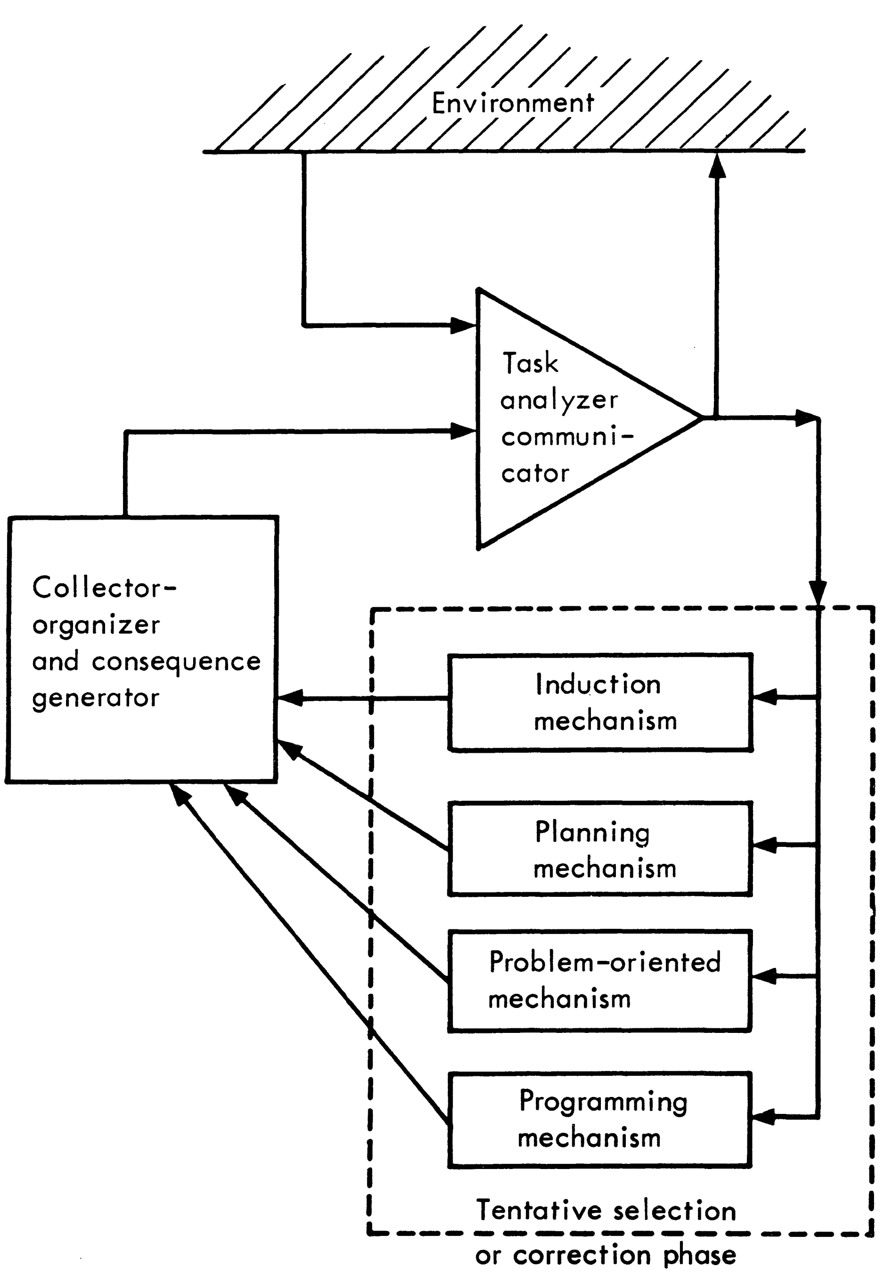


Figure 2. Master Feedback with Four Mechanisms Embedded (Mechanism Coordinator)

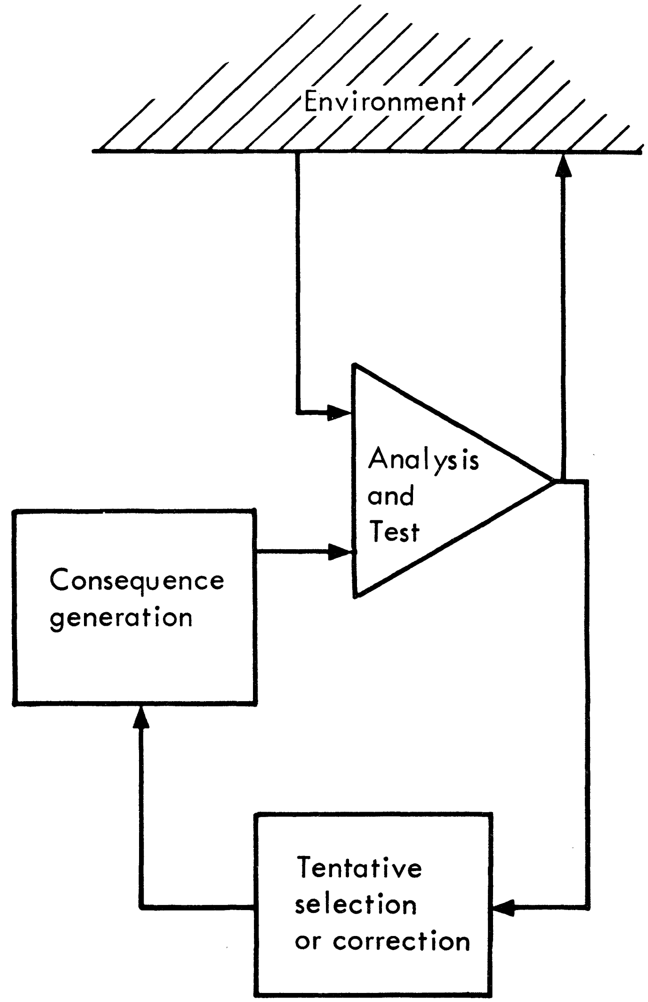


Figure 3. Basic 3-Phase Cycle Common to All

Figure 3 depicts schematically the basic three-phase cycle that is common to the mechanism coordinator and the four mechanisms. Figure 4 gives a closer look at the four mechanisms and shows the basic cycles, while suppressing the structure of the mechanism coordinator. The cycle in Figure 3, shown in general terms, passes through three phases: an analysis and test phase, a tentative selection or correction phase, and a consequence-generation phase. A feedback loop is formed when the analysis and test phase of the cycle receives the consequence-generation phase. Upon re-entering the analysis and test phase, reformulation or re-analysis of the given task is done by comparing the consequences with the description of the task. Selection of a new course of action or a modification of the previously proposed act is then effected in the tentative selection or correction phase. The three phases are repeated until a success or a failure is determined in the analysis and test phase.

This general structure is the basic framework used in the four special-purpose mechanisms and the mechanism coordinator. Their different functions and features are due to different sets of rules and restrictions about decision-making criteria, the environment, and the kind of information that is channeled through the loop. The environment of each of the four mechanisms is the mechanism coordinator, while the mechanism coordinator has as its environment, the Environment (the human user and natural or preprocessed input data). The specific function of each mechanism is generally implied by its name, as can be gathered from the following descriptions.

The programming mechanism is responsible for internal programming, i.e., internal generation of programs and associated data from externally given verbal descriptions or statements about what the programs are to do. The "units" of information that pass through the loop are basic operations and prestructured subroutines. Since a unit action, defined by the Environment for each task, is operationally defined in terms of subroutines and basic operations, this mechanism is said to handle "micro" actions below the level of observable behavior.

The problem-oriented mechanism is responsible for generating a sequence of unit actions leading to the solution of a given problem. Such unit actions are the units of information for this mechanism, and they are usually provided by the programming mechanism. Since the problem-oriented mechanism constructs and actually carries out the required sequence of legal moves in solving the given problem, this mechanism directly determines the behavior of the system. The problem-oriented mechanism is provided with procedural rules that help it to choose actions more efficiently than by exhaustive or random trial and error. These rules, however, are of a step-by-step nature, causing the mechanism to attack problems in piecemeal fashion and sometimes to lose sight of the total picture of the given problem. Consequently, correction of this "near sightedness" of the problem-oriented mechanism is attempted by the planning mechanism.

The function of the planning mechanism is to analyze the structure of a given problem and place guideposts on the road to the goal. The mechanism takes

a larger view, often at an abstract level, of a given task. After surveying the task as a whole, the planning mechanism subdivides the task into a hierarchy of subtasks, each presumably easier to perform than the original task. This hierarchy of subtasks constitutes a rough sketch of a possible course of action that guides the problem-oriented mechanism. The units of information here are state-descriptions, which represent stepping stones, or intermediate nodes, in the gap between two initially given nodes. Finding a complete solution can be thought of as establishing a valid path among many other possible paths in a tree of alternatives (Newell and Simon, 1964). In contrast, the solution plan suggested by the planning mechanism is incomplete, i.e., is made up of wider-spaced stepping stones. These stepping stones, or intermediate states, are handled by the mechanism coordinator, which instructs the problem-oriented mechanism to solve one portion of the problem at a time.

To make efficient use of its past experience, the problem-oriented mechanism is also influenced by the induction mechanism, which takes a still larger view of a given task, surveying the system's past experience with various problems and applying the experience to related problems not previously encountered. The units of information in this portion of the system are conjectures about a class of situations in terms of abstract patterns--patterns that, hopefully, reveal unifying principles or at least some similarity that will help in grouping things and events.

If we assume that the divisions in this system are reasonable divisions in the general structure of problem solving, a bright human seems to shine in the

phases of planning and induction. In a partnership between man and this system, man can, through the mechanism coordinator, give suggestions about a new conjecture to be tried or new subgoals to be set, and he can make requests for a printout of intermediate results from each mechanism (normal printout is only about unit actions).

A detailed account of Gaku's behavior in two specific problem-solving situations can be found in my article, "Gaku: An Artificial Student" (Hormann, 1965). Essentially, Gaku exhibited the ability to attack and finally solve a sequence of progressively more difficult problems, utilizing its previous experience and eventually finding a general solution pattern through the induction mechanism. The second problem-solving situation involved secondary learning: the human tutor gave suggestions and instructions about the kinds of generalizations Gaku could make or try out. Many refinements to Gaku are being added or investigated, hopefully to make the system general and powerful. These investigations are discussed separately in the following, since they are not necessarily unique to Gaku, but are important general questions raised by many who are concerned with machine intelligence.

Construction, utilization, and modification of a "cognitive map"

Throughout our lifetimes, we humans seem to build and modify very complex "cognitive maps," or internal models of the external world (the term was first introduced by E. C. Tolman). It is the utilization of such cognitive maps that enables us to internalize overt action. We often make judgments or decisions based on "what if" guesses, without actually experiencing the conditional events.

For well-defined problems such as chess, when we "look ahead" a few moves, the consequences of "what if" questions can be accurately answered. However, for most ill-defined problems that arise in real-life situations, the consequences of a proposed action must be estimated or approximated. As we gain more experience, we continue to modify our cognitive maps so that the consequences we generate hypothetically are accurate enough to serve in judging and making decisions effectively. In order for a machine to exhibit more than trivial intelligence, it seems imperative that it possess a similar ability. In Gaku, the consequence generation phase of the four mechanisms cannot work appropriately without the use of a cognitive map, however crude that map may be.

However, mere accumulation of information about the environment as a record of past experience will make such information increasingly inaccessible. Efficient utilization of a cognitive map depends on how the map is constructed and maintained up-to-date. A periodic "inventory" of the stored information should be provided that can result in transformation of information in the form of abstraction and generalization, and can lead to a better organization (by forming useful associations among blocks of information, including the techniques and methods applicable to them). Thus, the cognitive map can go through alternate phases of expansion (addition of more information) and contraction (by generalization and better organization). Of course, the difficult stages of generalization are those concerned with the discovery of unifying principles that tie things together in an useful way. We do not know much about how humans make generalizations--let alone how machines can.

Modification of the cognitive map should be effected by a program that analyzes feedback information. The modified cognitive map, in turn, would cause the behavior to change, and there must be a program to analyze the changed behavior along with the new feedback information. Therefore, it seems natural to assume that the cognitive map in the machine must include records of the machine's own behavior, perhaps even including records about <u>how</u> a particular choice of actions was made (in other words, it should know something about the function of its own mechanisms as well as the changes they cause). Of course, this makes the whole system very complicated. We will have to learn a lot from psychologists and philosophers, as well as from other areas of research--in particular, from simulation techniques. Simulation, in its simplified definition, is a model-building process. Dr. Greenberger (IFIP Congress, 1965) stresses the advantage of "dynamic" <u>formation</u> of the model (dynamic <u>execution</u> of the model is a usual feature in simulation). He thinks of the process of dynamic model formation as being carried out by humans, but parts of what humans do in this regard might be studied and "modeled" for cognitive map formation.

<u>A functional analogy to the human subconscious</u>.

Some psychological theories and explanations of how the human subconscious mind functions are detailed enough to be used in "molding the style of the machine mind." Our main concern is whether the incorporation of such features will result in a machine with desirable or interesting capabilities. In other words, we are not concerned with whether the resulting behavior <u>really</u> resembles the function of the human subconscious.

The following are some ideas that, when realized, may bear fruit.

1) Progressive grouping of different elements into larger and larger complexes. When we first learn to perform a sequence of actions with which we are unfamiliar, we consciously attend to each step of the sequence, but once we gain familiarity and confidence, we run through a sequence of actions without being consciously aware of each of its

Suppose we have acquired skills in performing a number of such sequences. When we attempt a complex task--which is an integrated sequence of those simpler skills--we need not attend to the full detail of each of the basic skills, but we attend to the way in which such "building blocks" are arranged. Depending on one's proficiency, the ultimate size of this grouping can be quite large. Many physical skills (e.g., piano playing and dancing) as well as mental skills (e.g., a child learning to read) seem to show this characteristic.

Suppose a machine has three modes of executing programs: interpretive, monitored, and unmonitored modes. In terms of programming techniques, the interpretive mode of operation means that the function of each instruction (usually at a higher level than the machine-instruction level) is translated into a set of instructions, which are then executed, keeping a record of the content and locations of the memory spaces involved. The monitored mode provides a means to check a program at crucial points to see that it is behaving in the intended way (e.g., at branch points or at the end, correlating the results with the task of the program). The unmonitored mode obviously has no

interruptions of the above kinds, and all the program instructions are executed at high speed in a predetermined order (the normal way of executing instructions in a computer).

Suppose a human describes a sequence of actions to the machine, which subsequently produces a sequence of machine instructions (one or more subroutines) to perform the described actions. Since there is no guarantee of correct performance (even an experienced human programmer has difficulty in ascertaining the workability of his program if it is sufficiently complex), the sequence will be executed interpretively and modified as necessary until a certain "confidence level" has been reached. This particular sequence of actions will then be performed in the monitored mode; in this mode, no "conscious" attention will be provoked unless some disturbances are reported at the critical points. Each "skill" thus developed will be identified with a symbol or a name, so that as the machine repertoire of such skills grows larger, more complex composite skills can be performed by calling the names of actions in a certain sequence, paying special attention to the way they fit together rather than to the detailed steps within each action. Speed of performance in such cases should be significantly greater than in cases where each action is interpreted or monitored.

2) Parallel processing of background information at the "subconscious" level. It is often evidenced that humans can, while consciously attending to a particular activity, pick up a piece of information (especially if it is something extremely desirable or undesirable) out of a vague background of

inputs from other modalities. For example, the driver of a speeding car, while talking intently with his passenger, catches a glimpse--out of the corner of his eye--of a black and white car; he immediately slows down, and simultaneously takes a second look--this time with full conscious attention--to decide whether or not it is indeed a police car. Three major characteristics are: (a) initially full attention was being paid to something else (the driver is not actively checking each car coming into his vision against the known attributes of a police car); (b) only a few simple cues (key attributes that signal either extremely desirable or extremely undesirable features) are necessary to provoke conscious attention; and (c) each situation may call for a different set of such cues (e.g.,if one were not driving, seeing a black and white car might have no effect on the direction of one's attention).

It is feasible to construct a combination of hardware devices now available that will manifest these characteristics (a, b, c, above) if crudely. The combination would consist of a computer, with its conventional memory, tied in a special way to an associative memory (or content-addressable memory). The main characteristic of the associative memory is that it possesses distributed logic--memory and computational functions are combined and distributed throughout the entire memory. Thus, logical operations can go on simultaneously in all cells of any selected part of the memory. A particular example is a search of the memory for registers containing a specified pattern of bits. In a conventional serial computer, this has to be done by bringing in the content of each separate register to the central logical processor where the actual comparison is made against the given paradigm (pattern of bits). Since an associative

memory can conduct such comparisons simultaneously, at the cells of the memory, the increase in speed over conventional processing can be many orders of magnitude, depending on the number of registers to be compared.

Figure 5 depicts schematically the interactions between the conventional computer and the associative memory. A stream of background information is shown coming in from E. (If E is the natural environment, the information must go through a preprocessor for digitization and some pattern recognition; if E is a human, he interprets the task environment and inputs preprocessed information.) As the information enters the associative memory, it is matched simultaneously against the contents of memory registers that represent the sets of desirable and undesirable cues for the given problem situation (prestored either by the human or by the conventional computer). If any of the cues match some parts of the input, that portion of the information enters the conventional memory by an action called "automatic interrupt" (different entry points are specified for both "desirable and "undesirable" interrupts). This automatic interrupt diverts the machine's "conscious" attention from whatever it has been doing, and causes the conventional computer to examine more closely the given pieces of information. The parts of the input that do not merit special attention (according to the sets of cues) are stored in the conventional memory for later examination.

In addition to input from E, the associative memory also receives input information generated in the conventional memory. In Figure 5, the curved arrow starting at the conventional memory and pointing to the associative memory

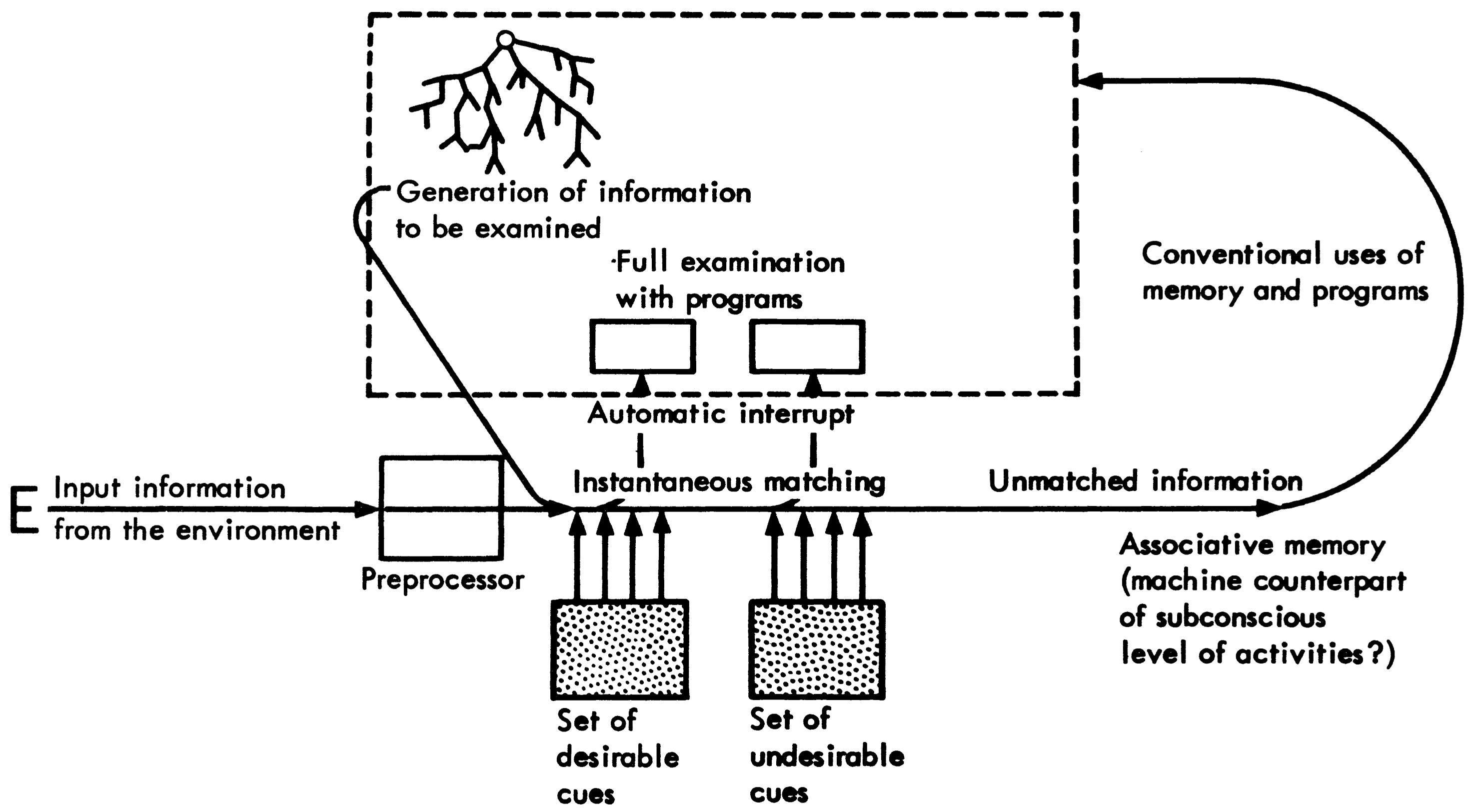


Figure 5. Interactions Between a Conventional Serial Computer and an Associative Memory

indicates this. In most problem-solving situations that are characterized by a high degree of symbol manipulation and hierarchical decision making, this flow of information is likely to be the dominant cause of the actuation of the associative memory. For example, a machine searching for a plausible sequence of steps for a problem solution may generate a large tree with many branches (alternative actions) and nodes (consequences of the actions as intermediate states). For complex problems, an exhaustive search, or even selective search with some "tree-pruning" heuristics, will exceed the practical limits of time and memory space. However, given an associative memory and sets of relevant cues, a large number of the consequences thus generated can be examined simultaneously for a quick estimate of fruitful avenues of exploration. In a game-playing situation, an experienced human player seems to use sets of such cues that tell him things to watch out for or things to take advantage of; he thus reduces to a more manageable size the amount of conscious examination he must expend.

In complex problem situations, more than one pair of cue sets (desirable-undesirable) is likely to be used. A system such as Gaku, for instance, might use one set for the problem-oriented mechanism in tree searching, another set for planning, and still another for induction. Also, a hierarchical organization of such pairs is possible within the associative memory, so that the pieces of information are matched against graded sets of cues or criteria; at different stages of problem solving, criteria for selection can be relaxed (like using a coarse sieve) or tightened (like using a fine sieve), depending on the purpose. (The resulting picture of the system combination will be much more complex than that depicted in Figure 5.)

Another example of the use of the associative memory is in planning. During the course of solving a single complex problem, one may become involved with a large assembly of interrelated subproblems. The ability to select apprcpriate subproblems that provide one or more plausible plans for the general problem solution often separates a good problem-solver from a poor one. There are no known general techniques or methods for effective selection, but an experienced and clever person seems to be able to make shrewd guesses, usually taking advantage of some peculiarities of the particular subproblems and associated methods.

In subproblem selection, as well as in the selection of fruitful directions of search in a game tree, such as the one mentioned above, humans use a rather inscrutable way of "applying relevant information." This process is very difficult to mechanize. The heart of the difficulty, it seems to me, is identifying relevant information--that is, criteria determination; we cannot use relevant information until we get it, and we cannot get it until we can say what it is. But this is where man functions best. In the man machine partnership, either man alone or a machine aided by a human can determine the sets of criteria for the given problem, and use of the associative memory can shrink the size of the search and speed up the desired information processing to a significant degree. Given this kind of efficiency, trying many different sets of cues for experimentation and future refinement can become feasible.

There still exist many problems in using an associative memory. One of them is how to program it, especially when it is connected to a conventional

memory, to ensure an optimal interaction between the two and to keep the associative memory in full operation. Another problem is choosing which block of information is to be stored in the associative memory and which is to be treated as the paradigm for comparison. Figure 5 shows one way. The reverse is to store the input information (either from E or from the conventional memory) in the associative memory and enter the sets of cues as paradigms for comparison. The choice will largely depend on the expected frequency of moving blocks of information in and out of the memory, and other features that influence efficiency and economy. At present, associative memories are expensive. However, as with automobiles, television sets--and even computers--increased demand will surely bring about more economical manufacture and lower costs. The associative memories of tomorrow need not be like those available today but they will have or include the functional characteristics with which I am concerned here, and hopefully they will be reasonable in cost and widely available.

There have been many suggested ways to take advantage of the features of the associative memory. However, the use of associative memory in conjunction with a conventional computer <u>in the manner</u> and <u>for the purposes</u> described in this section seems to be new. The technical feasibility of such a scheme has been confirmed by Mr. Paul Davies of Whittaker Corporation, well-known expert on associative memory and related fields. I am grateful to Mr. Davies for his valuable suggestions and comments.

3) The occurrence of sudden bright ideas. All of us have experienced the sudden "flash of insight" after mulling a problem over for fruitless hours. Thinking about how it happened and trying to figure out what favorable conditions contributed to the insight can give us the basis for some interesting speculations and some points of departure in trying to relate this process to machine intelligence.

This experience seems to have the following underlying characteristics (necessarily over-simplified): after a problem solver spends many hours on a problem with no solution in sight, he may set it aside and go on to other things (an incubation period after intensive and extensive study and effort). Detachment or a passive state of relaxation seems helpful, as if it promoted subconscious activity, for some time later, during a relaxed period in a seemingly unrelated context, suddenly a way to the solution (usually not the solution itself, but the end of a thread leading to the solution, or a picture of the whole rather than the details of the solution) comes to the problem-solver's consciousness.

Although it may seem that something came out of nothing, this "nothing" must really be a lot of things, e.g., the right combinations of right pieces out of thousands of possibilities. At the subconscious level--even while one is asleep--many activities, such as information mobilization, decomposition, and recombination, seem to be going on. Some seemingly unrelated pieces of information suddenly come together to form a meaningful whole by the addition of new information and/or the use of a new organization--or by finding a missing link.

The following is an interpretation--naive, to be sure--of such complex processes in terms of machine behavior. Suppose a sufficiently sophisticated adaptive machine, let us call it M, has attempted to solve many problems, succeeding in some cases and failing in others. M stores complete information about the previous efforts, including descriptions of the unsolved problems as well as of the solved ones. In the course of the new problem-solving situations, M acquires new methods, techniques, general principles, or new pieces of data. Suppose further that M is allowed, when it has no specific duty to perform or problem to solve, to go back to some of the old unsolved problems. If M is a system of programs on a time-shared computer, other people may be using the computer independently of M's activities (while M itself may have a number of users or human partners). Suppose it is arranged that whenever the total number of current users of the time-sharing system is less than N - n (N being the maximum capacity of the system and n some preset number less than N), then M is given an opportunity to "ruminate"--examining the old unsolved problems in the light of newly acquired skills and information. M may not succeed with any of them for many trials, but after many sessions of active learning and problem-solving, conditions may become ripe for arriving at the solution during one of the rumination periods (i.e., sufficient information, the necessary skills, and the right organization may be acquired).

If we further assume that the associative memory (as explained in the foregoing section) is used during the rumination period and a set of criteria for the desirable outcome (attributes of the goal states) was reset for the particular problem, then the automatic interrupt will trigger the attention of

the main computer, which will then check, fill in the final details, and print out the solution. Will the human user be surprised? He need not be, if he chooses to retrace the history of the solution process within the machine, for the solution depends on M's internal structure, its past experience (including its interaction with the environment), and the nature of the problem. In principle, the outcome is predictable. However, if the solution history is sufficiently complex, prediction will be difficult in practice.

For human insightful moments, no one really knows how and why only plausible or relevant combinations automatically come to the conscious level, while many unsuccessful or irrelevant ones (assuming that they are generated and examined at the unconscious level) remain below the surface. I am not offering an explanation, but the action of the automatic interrupt, connecting the associative memory to the conventional one, can offer a means to display a similar phenomenon in a machine. Aside from this interesting analogy, this type of combined system will have many practical uses.

Have an "ideating session" with your machine partner.

It may be that interaction with a responsive machine can open up new possibilities for stimulating and capturing man's creative activities. Suppose that there existed a machine of sufficiently high sophistication level, capable of carrying on a dialogue with the human user and stored with a body of existing information about a particular subject. The machine, when queried, could present the user with the possible relationships within that body of information and could call up blocks of substantive data as appropriate. Such a machine

might help the human user generate a multitude of ideas about new products, techniques, or problem-solving strategies, stimulating the man to ideate freely and extensively. As such an ideating session proceeded, one idea might spark another; some ideas might be modified, elaborated, or combined to form new ideas. If group ideation were preferred, a time-shared computer with many individual consoles would provide a number of participants with a convenient means to interact with the machine and with each other; each person would receive typed statements of ideas as they were offered (anonymity might contribute to less inhibited interaction by avoiding fear of ridicule or criticism), and could add his own ideas by typing them in. Simultaneous expression could go on without interference, at the same time allowing everything that anybody had to say to be available to the whole group (and thus stimulating even further ideation).

A singularly important role the machine could play here might not occur in the ideating session itself, but in the second phase of creative problem-solving, i.e., the evaluation phase. The volume of ideas and combinations of ideas generated during an ideating session might be overwhelmingly large; a human presented with so large a data base might not be able to make an effective evaluation--chains of consequences generated by all the ideas in all their combinations and permutations would defy effective analysis. This is where a machine partner with an associative memory could be of great help for simultaneous, selective information processing. The machine could collect, organize, transform, and select the available information, always in close

interaction with the man at crucial points of decision making. Together, they could determine criteria for selection of usable ideas and select methods of analysis (which could not usually be effectively determined in advance).

This type of machine-aided evaluation could also have psychological influence on the man during the ideation phase. Without such an aid, a problem-solver might easily be intimidated by the thought of having to make a few selections out of a large volume of ideas and alternatives (alternatives that might be diverse or even in conflict with one another); as a result, he might eliminate many explorable possibilities even before he started exploring. This kind of preselection is often observable in problem-solving situations of high combinatorial complexity. Of course, shrewd elimination of unfruitful paths may be what separates an expert from an average problem-solver, but everyone cannot be an expert, and wrong elimination can be costly. However, knowing that machine aid is available, even an average problem-solver may be prompted to be more explorative and experimental and to generate more ideas or alternatives (and their probable consequences) than he normally would. Ordinarily, quantity cannot replace quality, but we are considering the possibility here of <u>increasing</u> the quantity of ideas (good and bad) by a considerable magnitude. Even if novel and fruitful ideas were scantily and erratically distributed, there would tend to be more of them. The problem then becomes one of appropriate evaluation and selection.

In the following, I shall sketch a rough outline of some of the design features that could be included in a machine to act as a responsive partner in

ideating, and more important, in the evaluation part of creative problem-solving. Such a machine would not necessarily be either "intelligent" or "adaptive," at least in its rudimentary development stages, but it should be equipped with an efficient communication means that humans can easily learn to use. The current state of man/machine communication is still very restricted, but many research activities in user-oriented languages, English question-answering systems, and graphic input/output devices indicate that the future communication will be much easier. Furthermore, researchers in human creativity might observe the interactions of humans and a machine in a variety of disciplines--scientific, technical, and artistic; they could gather the comments of the users, and from this improve the machine design and find new ways to assist human creativity. Eventually, such systems might be used by scientists, managers, military commanders, inventors, writers, etc., who were working in problem situations that required inventive solutions.

Since each person's habits, skills, and purposes are different, a major concern of the machine design must be adjustability (to design changes) and flexibility (to individual needs). Such a machine would, however, probably incorporate some general patterns that have been discerned by researchers in human creativity. The user would be given suggestions about general steps to follow, but would be provided with a number of alternatives, including skipping some of the steps. Many of the ideas and techniques described in A Source Book for Creative Thinking (Parnes and Harding, Eds., 1962) and in Synectics (Gordon, 1961) are utilized here although they have been altered

considerably to fit the man-machine context.* For simplicity, the following description will be concentrated on one man, rather than a group of men, interacting with a machine.

A generally observed sequence of phases in complex problem solving consists of problem identification and definition, preparation and analysis of the problem, hypothesizing, synthesis, and testing or evaluation. In a complex task, this sequence, or a part of it, may be repeated once or many times. But in each phase of the sequence,an ideation-evaluation (I-E) technique can be used. For example, in the problem identification and definition phase the problem-solver may have only a vague idea about the problem situation when he begins the interaction. In such cases, either the user cannot define the problem at all, or the initially <u>assumed</u> problem may turn out to be quite different from the <u>real</u> problem. Therefore, it would not be a good idea for such a system to force the user to make a precise problem definition as an initial step. The primary function of the I-E technique in the first phase of problem solving would be to help the user arrive at a better understanding of what he wants and what the real problem is. Interaction with the machine in this phase might result in a set of plausible subproblems, forming one or more "plans of attack," or it might indicate the kind of information that should be investigated further. Sometimes ideas leading to the solution itself could be generated during this first phase; then the user could go directly to the synthesis or testing phase.

---

*Having a machine as an ideating aid, either in a group or alone, might obviate some of the criticisms of group "brainstorming"--that people very rarely <u>think</u> in groups, that a group can serve only to crush the individuality of its members,etc.

Similarly, the I-E technique can be applied to each of the other phases of the problem-solving sequence.

In the ideation part of the I-E technique, the machine's assistance could be limited to relatively simple "bookkeeping," except for a few prestored "suggestions" and some simple fact retrieval capability in order to supply the user with pieces of background information when requested. The problem-solver could use both symbolic input (words and phrases) and graphic input; with the proper technical developments, it might even be possible to include voice input, which would be transformed automatically into a printed form. The user would record his ideas just as they came from his mind.* The machine, in turn, would be programmed to type out suggestions that tended to stimulate further ideation and free association. For example, the machine might output lists of synonyms and antonyms of the user's words; other suggestions might include reminders to "list similar uses, functions, names, and objects," "think of what could exist as well as what does exist," "modify," "elaborate," "combine," "substitute," "find new and unusual relationships among_____, _____, . . . (here would appear a list of elements or words taken from the previously expressed ideas)," etc.

---

*In the ideation stage, the user is recommended not to exercise judgment on or criticism of ideas that emerge (however foolish or impractical they seem) in order to encourage free, uninhibited expression of thoughts and ideas. Sometimes "silly" or "wild" ideas tend to act as "spark plugs" in the generation of novel and fruitful ideas or they may themselves turn out to be novel and fruitful when placed in a different light or when other conditions are altered.

In a group ideating session with a machine mediator, however, the criticism of an idea may spark better ideas; the originator of an idea that is attacked may try to defend the idea; in doing so, he or other participants may come up with better ideas, or modifications that would remedy the idea's shortcomings.

If the problem-solver has his own favorite checklist to remind himself of certain ways of thinking or certain features to consider, he might give it to the machine at the beginning of the session. Furthermore, if the developers of the system could accumulate enough experimental data to make reasonable distinctions between different disciplines, the machine could be programmed to maintain several vocabulary sets, sets of background information, and sets of service routines (for the analysis and evaluation part of I-E) and the user could then indicate what problem category he was about to work on, activating a group of such data blocks and routines for use in his particular problem. After ideating for some time, the problem-solver would indicate to the machine that he was "thought out," or "run dry." He would then be urged to go on trying a little longer, for at this stage (according to researchers in human creativity) superficial blocks that dampen thought are often broken down by fatigue. This release from resistance allows free association between the conscious and the subconscious mind, and the results may be better ideas.

Finally, after the user had input everything he could in the ideation stage, the evaluation stage of I-E would begin. The kind of evaluation required would depend on the kind of problem, the thinking habits of the user in the ideation stage, and the particular problem-solving phase the session was currently in. It is impractical, therefore, to give any detailed steps to follow. In actual practice, the user might give the machine his ideas about the desired procedure, including some specific methods and techniques of analysis and evaluation. The machine could then compile these specifications

into a set of service routines that would be incorporated into the evaluation part of the I-E session, in addition to those capabilities the machine already possessed.

The following is a general, "uncustomized" kind of evaluation procedure. If the ideas that had been recorded in the ideation stage already represented many possible combinations of elementary ideas or components, the user could start the evaluation procedure from step 2; if they did not appear to cover possible combinations sufficiently, it would be recommended that the user start from step 1.

Step 1. In this step, deliberate decomposition of the ideas could be tried, so that a systematic recombination could be done by the machine to generate more possibilities. An elaborate and systematized technique of decomposition and recombination has been introduced by Dr. F. Zwicky. The technique, called "Morphological Analysis," is explained in A Source Book for Creative Thinking, and for lack of space the explanation will not be repeated here; essentially, it is a technique of decomposition by identifying independent variables and their components, for the purpose of making subsequent recombination "meaningful." The book does not mention any assistance from a computer, but the large number of combinations usually produced by this method naturally suggests the use of a computer.

After the morphological analysis had identified a number of independent variables and their components (not necessarily completely or accurately at the first try), the machine would be asked to generate all the possible combinations of variables. Then the next step, step 2, would be followed.

<u>Step 2.</u> In this step, the user would describe what he was looking for--would establish sets of desired and undesired properties or attributes. This process--the determination of appropriate sets of criteria for selection--is where human ingenuity must be fully exercised. The user might determine desirable attributes with a rank order, so that the machine would make successively fewer and more restricted selections; he might specify undesirable features, so that the machine could delete worthless combinations and he could examine what was left over; he might define a set of attributes to select the "normal" or "usual," and examine what was left over (an inventor might be interested in something "unusual," which itself could not be defined).

<u>Step 3.</u> The user would examine and describe each combination in Step 1 that had been generated in terms of its attributes, at the same time discarding some obviously unusable ones. The attributes should be determined with the problem's total solution in mind, and should include both "desired" and "undesired" features.

The user might assign attributes to the components identified in step 1 <u>before</u> they were combined, thus reducing the number of items he needed to examine. Of course, this method would not apply to cases in which the attributes of components in combination differed from the attributes of those components as individuals.

<u>Step 4.</u> The machine would then match the attributes of each combination (assigned in step 3) with the set of criteria for selection (established in

step 2), and the selected combinations could be output for the user to examine further. This kind of machine assistance would become more valuable as the number of combinations increased and the attributes became more numerous. If there were only a few combinations and a few attributes for each, then step 3 itself could provide immediate selection. However, after ideation, the selection is likely to be too complex and unwieldy for straight comparison and selection by the human user.

Step 5. Depending on the results in step 4, all or parts of the previous steps could be repeated. As the problem-solver experimented with different selection methods and criteria and compared their results, he would be likely to gain a deeper insight into the problem situation and into what he was really looking for; e.g., the independent variables and their components tentatively defined in step 1 could be revised.

The important conditions to be maintained here are flexibility and a spirit of experimentation; the user should not feel "committed" to his initial attempts to describe either the problem itself or its criteria completely and accurately. "Wild guesses" should be welcomed and experimentation encouraged. If the associative memory were being used in machine selection of items, the man would not need to wait too long to see the consequences of his criteria determination, and might be prompted to have second and third thoughts for further experimentation and refinement. Some statistical analysis on the accumulated data, if suitable, might then be requested of the machine.

<u>Mt. Everest with rope and pitons, or Mt. Fuji in a jeep?</u>

Some problems can be attacked either the hard way or the easy way; some problems just do not offer any easy way out. Even if we find an easy way, it may or may not be a <u>better</u> way, depending on the objective. In the foregoing sections, I have pointed out the difficulty of understanding how a clever problem-solver chooses "relevant" information out of many possible sources and uses it "appropriately," and I have suggested that the man in a man-machine partnership can determine and supply the criteria so that a machine will have something on which to base its selection decisions.

From the point of view of "artificial intelligence," this is an easy way out, or even downright cheating; the main purpose of artificial intelligence research is to construct a machine that can exhibit many aspects of intelligence, and the ability to detect relevance and determine criteria is certainly one of these aspects (and an important one indeed). The problem of using relevant information is half solved when the criteria for relevance are supplied by the human user. I am, in a sense, defending both the easy and the hard way; I believe the easy way does not substitute for the hard way, but leads us toward better understanding of the problems that make the hard way hard.* We

*Besides, the "easy way out" is easier only in a relative sense; it does not solve everything. For example, how do we determine a "product with best marketing potentials"? We will probably need a simulation model of relevant marketing behavior; if the product in question is unlike anything that has been on the market, however, prediction will be very difficult. In such a case, a straight listing of very general criteria for market receptivity would do just as well. Another problem is to express relevance that is only discernible in terms of relational concepts (criteria for selection that are expressible in terms of conjunction and/or disjunction of attributes are relatively easy to use).

do not know much about the mechanisms involved in discerning relevance. However, in the process of assisting the machine to determine criteria at crucial points in problem-solving processes, man will be forced to explicate his intuitive thinking and may gain a deeper understanding of the mechanism (his own) and the process of criteria determination.

Such an "easy way out" can also be used to investigate other related concepts, e.g., different uses of the concept of similarity (or analogy) in different problem-solving processes. The concept of similarity, seemingly related to the concept of relevance, is involved in many troublesome areas of artificial intelligence, such as induction, generalization, planning, and the development of the cognitive map. As in the case of "using relevant information," using what the problem-solver knows about "similar" problems, methods, techniques, principles, etc., begins with determining criteria for similarity. However, the criteria of similarity are just as elusive as the criteria of relevance. We seem to use the same term "similar" quite dissimilarly in different contexts.

It is rather basic and common practice for us humans that, when we face a new problem, we first try to use methods similar to those that have worked in the past on similar problems. We want our machines, too, to be able to utilize their past experience. But what do we mean by "similar" problems? Can we detect similarity from the problem statements? Some problem statements that appear similar may call for quite different solution methods and techniques, and problem statements that appear very different on the surface may turn out to yield to similar solution methods. Therefore, grouping problems together

as "similar" must be done in a useful sense--some "relevant characteristics" of a given problem must be discerned that eventually point to similar methods of attack. The troubles in the above situation will be doubled if we do not know what we mean by "similar" methods--but do we?

This elusive use of similarity judgment is one of the causes of difficulty in mechanizing problem-solving techniques--techniques for planning (generation and judicious selection of subproblems), for estimating the difficulty of, and alloting effort to, the subproblems (in order to avoid chasing indefinitely down blind alleys and to initiate other possible avenues of exploration), for recognizing partial success (when a particular approach fails, seldom is the total effort wasted), for generalizing past experience to encompass problems and situations the machine has not encountered (induction, abstract pattern recognition, etc., enter this process), and for developing cognitive maps. If some of these show no obvious use of the concept of "similarity," I suggest that the reader try to find different ways in which the concept of similarity (in both degree and kind) creeps in.

When we say "similar objects" we usually mean that some attributes of the objects have the same or close values, but the rank order of "relevant" attributes seems to change with contexts; we observe that two similar objects are no longer to be considered similar when we find them in a different situation. The term "similar pattern" stresses the <u>ways</u> in which component objects are related within composite objects. This in turn raises the question of what we mean by "similar relations." "Similar situations" seems to indicate a

similarity within whole collections of data--both events and objects--but we must first know what we mean by "similar events." When we say "similar transformations," we sometimes mean that the results of applying transformations are similar, and other times the transformations themselves. The root of these troubles may be the difficulty of representing abstract concepts of qualities quantitatively by the use of discrete symbols and values, which in turn are interpreted by the machine according to a certain predetermined sets of rules. Attributes and their values, no matter how finely detailed, are still inadequate and imperfect representations of a vaster "something" lying behind them. Humans are capable of "feeling" or "sensing" this something, and so of "filling in" things not explicitly expressed.

Thus, the "easy way out," though it does not solve the real problem, can provide a means to combine man's intuitive ability with a "responsive" machine in a wide variety of applications. It is not feasible to prestore in the machine all the possible contexts or situations (and the corresponding sets of rules governing similarity, relevance, etc.), some of which are not even conceived at the time the machine is designed. When an easier way is available, why not use it to explore more areas of possibility? By this path, we may eventually reach real heights in the molding of a "machine mind."

REFERENCES

Armer, P. Attitudes toward intelligent machines. Datamation, 1963, 9(3), pp. 34-38; 1963, 9(4), pp. 29-32.

Borko, H. (Ed.) Computer applications in the behavioral sciences. Englewood Cliffs: Prentice-Hall, 1962.

Davies, P. M. Design for an associative computer. Proceedings of the 1963 Computer Conference, IEEE, T-147, pp. 109-117.

Duncker, K. On problem-solving. Lynne S. Lees (Tr.) Psychological Monographs, 1945, 58(5), Whole No. 270.

Feigenbaum, E. A. and Feldman, J. (Eds.) Computers and thought. New York: McGraw-Hill Book Company, Inc., 1963.

Gordon, W. J. J. Synectics. New York: Harper & Brothers, 1961.

Greenberger, M. A general purpose system for the structuring on-line of information and decision processes. Information Processing 1965, Proceedings of IFIP Congress 65, v.2. (In press.)

Hormann, A. M. Programs for machine learning, Part I. Information and Control, 1962, 5(4), pp. 347-367.

Hormann, A. M. Programs for machine learning, Part II. Information and Control, 1964, 7(1), pp. 55-77.

Hormann, A. M. Gaku: An artificial student. Behavioral Science, 1965, 10(1), pp. 88-107.

Kelly, J. L., Jr. and Selfridge, O. G. Sophistication in computers: A disagreement. IRE Trans. Information Theory, 1962, IT-8, pp. 78-80.

Kochen, M., MacKay, D. M., Maron, M. E., Scriven, M. and Uhr, L. Computers and comprehension. The Rand Corporation, Santa Monica, California, Memorandum RM-4065-PR, April 1964.

Licklider, J. C. R. and Clark, W. E. On-line man-computer communication. Joint Computer Conference Proceedings, v.21 (Spring), May 1962, pp. 113-128.

Licklider, J. C. R. Man-computer partnership. International Science and Technology, May 1965, pp. 18-26.

Licklider, J. C. R. Man-machine interaction: Remote consoles and displays. Information Processing 1965, Proceedings of IFIP Congress 65, v.2. (In press.)

MacKay, D. M. The epistemological problem for automata. In C.E. Shannon and J. McCarthy (Eds.), Automata studies. Princeton: Princeton University Press, 1956, pp. 235-251.

McCarthy, J. Programs with common sense. Mechanisation of thought processes, v.1, National Physical Laboratory Symposium No. 10. London: Her Majesty's Stationery Office, 1959, pp. 75-84.

Miller, G. A., Galenter, E., and Pribam, K. H. Plans and the structure of behavior. New York: Henry Holt & Company, 1960.

Minsky, M. Learning systems and artificial intelligence. Applications of logic to advanced digital computer programming. Ann Arbor: University of Michigan, College of Engineering, 1957.

Minsky, M. Steps toward artificial intelligence. Proc. IRE, 1961, 49, pp. 8-30.

Minsky, M. Descriptive languages and problem solving. Proc. WJCC, 1961, 19, pp. 215-218.

Montalbano, M. S. Notation, insight, and problem definition methods. Information Processing 1965, Proceedings of IFIP Congress 65, v.2. (In press.)

Newell, A., Shaw, J. C., and Simon, H. A. A general problem-solving program for a computer. Computers and Automation, 1959, 8, pp. 10-17.

Newell, A., Shaw, J. C., and Simon, H. A. A variety of intelligent learning in a general problem solver. In M. C. Yovits and S. Cameron (Eds.), Self-organizing systems. London: Pergamon Press, 1960, pp. 153-189.

Newell, A. Some problems of basic organization in problem-solving programs. In M. C. Yovits, G. T. Jacobi, and G. D. Goldstein (Eds.), Self-organizing systems. Washington, D.C.: Spartan Books, 1962, pp. 393-423.

Newell, A. and Simon, H. A. Problem-solving machines. International Science and Technology, December, 1964, pp. 48-62.

Newell, A. The search for generality. Information Processing 1965, Proceedings of IFIP Congress 65, v.1. Washington, D.C.: Spartan Books, 1965, pp. 17-24.

Oettinger, A. G. Automatic processing of natural and formal languages. Information Processing 1965, Proceedings of IFIP 65, v.1. Washington, D.C.: Spartan Books, 1965, pp. 9-16.

Parnes, S. J. and Harding, H. F. (Eds.) A source book for creative thinking. New York: Charles Scribner's Sons, 1962.

Reitman, W. R. Information-processing models in psychology. Science, 1965, 144, pp. 1192-1198.

Reitman, W. R. Programming intelligent problem solvers. IRE Transactions on Human Factors in Electronics, 1961, HFE-2, pp. 26-33.

Sass, M. A., and Wildinson, W. D. (Eds.) Computer augmentation of human reasoning. Washington, D.C.: Spartan Books, Inc., 1965.

Selfridge, O. G. Pandemonium: A paradigm for learning. Mechanisation of thought processes, v.1, National Physical Laboratory Symposium No. 10. London: Her Majesty's Stationery Office, 1959, pp. 511-531.

Simon, H. A. Simulation of human thinking. M. Greenberger (Ed.) Management and the computer of the future. Cambridge: MIT Press, 1962, pp. 94-131.

Simon, H. A. How computers can learn from experience. In W. F. Freiberger and W. Prager (Eds.), Applications of digital computers. Boston: Ginn and Company, 1963, pp. 11-27.

Sutherland, I. E. The ultimate display. Information Processing 1965, Proceedings of IFIP Congress 65, v.2. (In press.)

Sutherland, I. E. On-line graphic systems--an overview. Information Processing 1965, Proceedings of IFIP Congress 65, v.2. (In press.)

Thomson, R. The psychology of thinking. Baltimore: Penguin Books, 1959.