The Learning Research and Development Center of the University of Wisconsin is engaged in a long-term multifacet study of concept learning, supported by the U.S. Office of Education. The Concept Attainment Simulation Experiment (CASE) is the facet of this overall effort which utilizes the technology of computer simulation as a vehicle for obtaining a better understanding of the psychological processes involved in the learning of concepts. The long-range goal is the utilization of the insights thus obtained to improve classroom learning. The study of concept learning has a long history within psychology and has received considerable attention in recent years due in part to the book by Bruner, Goodnow, and Austin which delineated strategies for learning concepts. The experimental materials used by Bruner consisted of a finite universe of objects each of which possessed \( n \) dimensions; and each dimension could assume \( k \) different values. A classification rule (a concept), consisting of a particular combination of dimension values, partitioned the universe into two mutually exclusive sets. In a typical experiment a subject was shown an object which was an exemplar of the set defined by the concept and told his task was to ascertain the classification rule. In order to attain the concept the subject chose objects from the universe and the experimenter indicated the set membership of the object chosen. The object selection-designation procedure continued until the subject could verbalize the correct classification rule and hence the concept had been attained. The experimental situation, the problem to be solved, and learning procedure involved appear reasonably simple and a number of persons have written programs to simulate this type learning experiment—Hunt and Hovland,\(^2,3\) Hunt,\(^4\) Allen,\(^5\) Wickelgren,\(^6\) and Baker. The book by Hunt provides an excellent review of much of the psychological literature relevant to concept learning as well as a discussion of his own simulation program. Unfortunately the existing programs leave one with the disquieting feeling that although they attain concepts, little has been added to our understanding of the psychological processes involved in concept learning. Most of these programs are at best watered-down algorithms and involve very little of psychological importance. Because of the shortcomings of the existing simulation programs a project was initiated to develop a program which hopefully will eventuate in something of psychological significance.

The basic approach was to use Bruner's notions about learning strategies, coupled with concepts regarding the structure of behavior from the book by
Miller, Galanter, and Pribam to write a computer program which would attain concepts. This initial program based on semi-theoretical grounds would then serve as a stepping-off point for a learning process on the part of the present author.

A system for collecting data was established which consisted of a closed feedback loop, with the simulation program at one end and protocol gathering during experiments involving human learning at the other end. Within the computer program, certain routines may be based upon a priori grounds or represent areas not clearly understood. In order to get better insights into such areas, questions are used during the protocol gathering which will elicit verbalizations relevant to those points. Thus, the computer program guides the production of information within the protocol which is subsequently used to modify the program itself. By making an extremely close connection between the computer program development and the learning experiments with human subjects the hope is to obtain a better understanding of the psychological processes involved. Having set the broad context within which the project operates, let us next turn our attention to the actual computer program involved.

THE CASE PROGRAM

Memory Structure

During the early phases in the evolution of the CASE program it became obvious that one of the keys to the problem was an adequate representation of the structure of human memory. The psychological literature contains a considerable body of material related to memory and much of this was studied to ascertain an appropriate structural form of memory. The result of this search was to design a memory consisting of three levels: Working memory (WM), short-term memory (STM), and long-term memory (LTM). The working memory is a unit which serves two functions. First, it holds all information received from the external environment until it can be analyzed and re-coded for transmission to a more permanent level of memory. Second, it serves as a buffer memory for holding information which is created within the subject and must be passed from one information-processing routine to another. In this buffer mode it provides certain higher-level routines with contextual information which is used to guide program flow. The short-term memory is semipermanent and retains information relevant to the current state in the learning of a particular concept. Short-term memory can receive inputs only from routines which re-code and transmit the contents of working memory or long-term memory. Long-term memory will contain information re-coded from short-term memory concerning concepts learned and how they were learned but at the present time only working memory and short-term memory have been programmed. Figure 1 illustrates the communication paths within the memory structure. The only means of communication from STM and LTM to the external world is via the output channel. For example, the subject tells the experimenter which object he has selected via this channel but the experimenters designation of the set membership of the object is received by the subject via working memory.

The internal structure of short-term memory consists of lists having a somewhat unusual IPL-V structure which has proven extremely useful. The
structure employs two levels of attributes; the class attributes which represent a rather broad description such as the permanent characteristics of an object; and specific attributes such as an object's serial position in the external environment, thus providing a detailed description within the class attribute. Table 1 illustrates a typical list within STM. The description list 9-0 describes M13, des-

Table 1. Typical memory list in short-term memory.

<table>
<thead>
<tr>
<th>M10 9-0</th>
<th>E0</th>
<th>E1</th>
<th>E2 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>9-0 0</td>
<td>E0 9-1</td>
<td>A1X1</td>
<td>V1 0 X2 0</td>
</tr>
<tr>
<td>9-1 0</td>
<td>A2</td>
<td>V2</td>
<td></td>
</tr>
<tr>
<td>V2 9-2</td>
<td>Y2</td>
<td>Y1 0</td>
<td></td>
</tr>
<tr>
<td>9-2 0</td>
<td>A3 Y2 9-3 0</td>
<td>V3 0 9-3 0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>A4</td>
<td>A5 V5 0</td>
<td></td>
</tr>
</tbody>
</table>

cription list 9-1 describes the symbol E0 on M13. The description list 9-1 contains class attributes, such as A2, and its attribute value list V2, which is merely a storage device for symbols whose function is to hold a description list containing the specific attributes and their values (A4, V4: A5, V5). The value list of the class attribute is a push-down list whose top symbol always represents current information.

Notice in Table 1 that the list structure is symmetrical in form to the upper left and lower right of the dotted line. The symmetry enables one to write simple routines which function for a module of memory regardless of the level at which the module occurs within the structure. There are four such basic memory routines which do all of the STM input and output:

1. Remember a name.
2. Remember something about that which has been named.
3. Recall a name.
4. Recall something about that which has been named.

At the present time we have not attempted to include LTM or to introduce forgetting or interference; however, we anticipate at some point building such mechanisms into the memory structure.

Program Structure

The CASE computer program has been designed with an expandable hierarchical structure whose depth depends upon the level of sophistication obtained in understanding the learning process. At the present time there are four levels, with each level being tested within the next higher level as an IPL-V list structure. The list structure representing the learning process is presented as input to a special interpreter, which executes the symbols in the structure and performs a number of housekeeping functions. The upper level (S) specifies what Bruner et al. refer to as a strategy, and is a list of symbols which represent major procedures within a strategy. The next lower level is the procedure level (Z-D) which is a list of symbols representing the processes combined to accomplish a given procedure, such as searching the external environment for an object having certain characteristics. The next lower level (P-Q) consists of information-processing routines written in IPL-V and is the lowest level that the program can manipulate at run time. The fourth level (R) consists of basic information processing modules coded in IPL-V which a programmer can use to manually write new P-Q level routines. The R's are subroutines which do such things as compare, test for the presence or absence of information, etc. Within each level it is necessary to maintain a sharp distinction between routines which perform operations (Z's and P's) and those which provide decision-making information (D's and Q's). It has also been found necessary to defer decision making upward to the next higher level for action. It should be noted, that all levels above the P-Q and R levels merely consist of symbols, hence one writes IPL-V only at the lowest level—a fact which has many implications for how one studies the learning process.
In that the P-Q level routines are basic information-processing routines, they can be used in a wide variety of situations within the program where the processing is identical but the information, its source, and its disposition differ. Because of this characteristic of the P-Q level routines, one is faced with the problem of how to use the same routine in a number of different contexts without developing a significant amount of situationally dependent IPL-V coding. The solution devised uses a pseudo code whose description list contains the inputs, outputs, and characteristics of the routine. The inputs to a routine are determined by a higher-level routine called a contexter which in the case of the P-Q level routines uses the contents of working memory to determine what the inputs to the processing routine should be for a given situation. Currently the outputs are normally placed in working memory although other options are possible. Although we have not done so yet it appears feasible to put on the description list of the routine a specification of the kind of information processing the routine is capable of performing. Table 2 shows a P level routine before and after the context program has functioned. Once the inputs have been determined from the context, the routine is returned to the interpreter for execution. Although it has not been done, the concept of context routines which operate at all levels within the program structure appears feasible and the context routine at the highest level would be a plan to create plans such as suggested by Miller, Galanter and Pribam.8

Table 2. Pseudo Code System.

<table>
<thead>
<tr>
<th>Before Context Program</th>
<th>After Context Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>P31 9-0</td>
<td>P31 9-0</td>
</tr>
<tr>
<td>P30 0</td>
<td>P30 0</td>
</tr>
<tr>
<td>9-0</td>
<td>9-0</td>
</tr>
<tr>
<td>A7</td>
<td>A7</td>
</tr>
<tr>
<td>V7 0 0</td>
<td>V7 M11</td>
</tr>
<tr>
<td>A8</td>
<td>A8</td>
</tr>
<tr>
<td>V7 M1 0</td>
<td>V7 M1 0</td>
</tr>
</tbody>
</table>

Fundamentally the program is still an algorithm as it very efficiently learns every concept attempted in a minimum number of object choices. Although it currently shares this fault with its published predecessors, we feel its internal structure is more psychologically oriented and the potential for nonalgorithmic behavior exists.

SOME RETROSPECTS

Program Characteristics

When one reviews the history of the CASE program it becomes quite clear that a subtle process is in effect. Namely, as one's understanding of the learning process increases, the computer simulation program changes from routines which perform a
large block of the concept attainment process to a number of short routines which can be widely employed. In the CASE program such a change has been dramatic at the P-Q level from Mark 1 Mod zero to Mark 3 Mod 1. The cynic will counter that we are merely learning how to code IPL-V but I do not believe this is the only basis, as the change has been effected primarily on psychological grounds rather than coding considerations. In fact, separate symbols are used to designate routines which are the result of IPL-V rather than psychological considerations and the former routines are quite rare. The character of the subroutines in the CASE program have also changed from being highly specific to the Bruner type experimental situation to being reasonably independent of the experimental situation. They are, however, dependent upon the basic memory structures defined earlier. The situationally dependent tasks still get performed but the computer program is problem specific at a higher level than was previously true.

Outcomes of the Case Program

There have been a number of outcomes of the CASE effort which are as follows:

1. The hierarchial structure of the processing routines and what appears to be a parallel structure in context routines has led us into a continual search for logical units within the learning process. Behaviors which once seemed quite dissimilar have been decomposed and found to share a number of basic information processing modules. As a result we are slowly acquiring a better understanding of the psychological processes involved in concept learning.

2. The development of the CASE program has generated ideas for classical psychological experiments in a number of areas as a result of problems arising during the development of certain subroutines. Topics such as the role of dominant dimensions in a subject’s learning behavior and the lack of independence among dimensions in the Bruner experimental materials have been elicited. It appears as though an important outcome of computer simulation is the generation of ideas which can be researched in the usual psychological experimental setting.

3. A completely unexpected outcome has been that we rarely make a production run on the computer, a fact which seems anomalous in a computer simulation project. What has happened is that enormous numbers of man hours have been devoted to gathering and studying protocols, to the development of programming techniques in order to implement the next level of sophistication within the system, and to analysis of the computer program itself. These activities plus the lack of variability in the learning behavior of the CASE program at this point in time have resulted in a relatively few production runs.

CONCLUSIONS

The CASE project has not been conceived as an effort which will produce immediate spectacular results, rather we view it as a slow developmental process. The memory structure, the program structure, the interpreter, and the contexter are basic concepts which we feel will enable us to continuously improve the sophistication of the program as our understanding of the concept learning process improves. Although it is difficult to single out specific accomplishments of great psychological importance, a certain modicum of progress has been made along these lines. At the current time the computer program is primarily a medium for expressing and storing the insights and understandings of the concept learning process which we have acquired. Uhr has previously indicated that psychological theories might be expressed in the form of computer programs and our experience to date tends to substantiate this point of view.

REFERENCES


