DOCUMENT RESUME

ED 112 951 IR 002 676

AUTHOR Beard, Marian; And Others
TITLE The Improvement and Individualization of Computer-Assisted Instruction: Final Report.
INSTITUTION Stanford Univ., Calif. Inst. for Mathematical Studies in Social Science.
SPONS AGENCY Advanced Research Projects Agency (DOD), Washington, D.C.; Office of Naval Research, Washington, D.C. Personnel and Training Research Programs Office.
PUB DATE 15 Sep 75
NOTE 42p.

EDRS PRICE MF-$0.76 HC-$1.95 Plus Postage
DESCRIPTORS College Students; *Computer Assisted Instruction; Educational Strategies; *Individualized Instruction; Individualized Programs; Instructional Design; Instructional Technology; Models; Program Descriptions; Programed Materials; Programed Tutoring; Program Evaluation; *Programing; Second Language Learning; Vocabulary Skills
IDENTIFIERS Algebraic Interpretive Dialogue; BASIC Instructional Program; Optimization of Learning

ABSTRACT

Four research projects were conducted on the improvement of individualized instruction. Two methods of teaching foreign language were examined. In the first, the computer stored a profile of the student's previous performance in German vocabulary, and it developed a strategy to teach the student additional German words. The second project tested the effectiveness of a special keyword association method to maximize the retention of Spanish and Russian vocabulary. The second two projects were computer-assisted courses in computer programing: Algebraic Interpretive Dialogue (AID), and BASIC Instructional Program (BIP). In both, the computer combines the student's history and the structure of the curriculum to construct the optimal teaching strategy. (EMH)

THE IMPROVEMENT AND INDIVIDUALIZATION OF COMPUTER-ASSISTED INSTRUCTION: FINAL REPORT

by

Marian Beard, Avron Barr, Dexter Fletcher, and Richard C. Atkinson

This research was supported jointly by:

Office of Naval Research
Psychological Sciences Division
Personnel and Training Research Programs (Code 458)
Contract Authority Number: NR 154-326
Scientific Officers: Dr. Marshall Farr and Dr. Joseph Young

and

Advanced Research Projects Agency
ARPA Order Number: 2284 dated 30 August 1972
Program Code Number: 3D20

Contract Number:

N00014-67-A-0012-0054
1 August 1970 - 31 July 1975

Principal Investigator:

Richard C. Atkinson
Professor of Psychology
Institute for Mathematical Studies in the Social Sciences
Stanford University
Stanford, California 94305
(415) 497-4117

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1 REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) The Improvement and Individualization of Computer-assisted Instruction: Final Report		5. TYPE OF REPORT & PERIOD COVERED Final Report August 1970-July 1975 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Marian Beard, Avron Barr, Dexter Fletcher, and Richard C. Atkinson		8. CONTRACT OR GRANT NUMBER(s) N00014-67-A-0012-0054
9. PERFORMING ORGANIZATION NAME AND ADDRESS Institute for Mathematical Studies in the Social Sciences - Stanford University Stanford, California 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N RR 042-0; RR 042-0-0 NR 154-326
11. CONTROLLING OFFICE NAME AND ADDRESS Personnel & Training Research Programs Office of Naval Research (Code 458) Arlington, VA 22217		12. REPORT DATE September 15, 1975 13. NUMBER OF PAGES 33
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15 SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

computer-assisted instruction (CAI), instruction control strategy, instructional theory, optimized learning, tutorial CAI, second-language vocabulary learning, tutorial programming laboratory, BASIC Instructional Program (BIP), Algebraic Interpretive Dialogue (AID)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report summarizes research on the improvement and individualization of instruction, with reference to a theoretical framework of optimization, and specific applications in computer-assisted instruction (CAI). Two projects involved the acquisition of foreign-language vocabulary items. The first (using German vocabulary) concerned itself with optimizing the selection of items for study, where the optimization criterion was the number of items retained on a posttest. A second vocabulary acquisition project concerned

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

the application of a mnemonic technique called the keyword method. Experiments with Spanish and Russian items showed that the method could be a powerful aid in building and retaining a large vocabulary of foreign words.

Two CAI courses in computer programming were developed. The first provided instruction and practice in the Algebraic Interpretive Dialogue (AID) language, and was used to investigate various optimization and individualization techniques, on the level of single problems as well as over entire lessons. The BASIC Instructional Program (BIP) was developed as a vehicle for CAI research in optimal selection of instructional material by means of an internally stored curriculum description and a model reflecting the student's changing state of knowledge. Both the student history and the curriculum organization are used to select problems in a dynamic way, as each is directly related to the content of the course, described as sets of specific programming skills.

Each project is discussed briefly, with references to the publications that describe the research in detail.

SUMMARY

This report summarizes the research conducted at the Institute for Mathematical Studies in the Social Sciences by Professor Richard C. Atkinson and his staff on ONR Contract No. N00014-67-A-0012-0054, August 1970 to July 1975. The central theme of the research is the improvement of instruction, with reference to a theoretical framework of optimization of the learning process, and specific applications in computer-assisted instruction (CAI).

A theory of instruction is measured against the following criteria: (1) a model of the learning process; (2) specification of admissible instructional actions; (3) specification of instructional objectives; (4) a measurement scale that permits costs to be assigned to each of the instructional actions and payoffs to the achievement of instructional objectives. To the extent that these four criteria can be formulated explicitly, optimal instructional strategies can be derived. Four projects, in two distinct subject areas, were carried out with the purpose of applying and extending the concept of a theory of instruction.

Two projects involve the acquisition of foreign-language vocabulary items. The first (using German vocabulary) concerned itself with optimizing the computer-controlled selection of items for study, where the optimization criterion was the number of items retained on a posttest. The optimal strategy developed was based on a mathematical model of vocabulary learning; the model is used to compute, on a trial-by-trial basis, an individual's current state of learning. Based on these computations, items were selected to optimize the level of learning achieved at the end of the instructional session.

A second vocabulary acquisition project concerned the development and application of a mnemonic technique called the keyword method. The technique requires the student to construct a chain of two links between the foreign word and its English translation. An audio link connects the foreign word to an English word with a similar sound (the keyword); an imagery link connects the keyword to the English translation by means of a strong visual image. Experiments with Spanish and Russian vocabulary items showed that the method could be a powerful aid both in building a large vocabulary in a short time and in retaining the material for recall on delayed tests.

Considerable research effort was devoted to two CAI courses in computer programming. The first course provided instruction and practice in the Algebraic Interpretive Dialogue (AID) language. The AID course was an attempt to apply instructional theory to a full scale curriculum, as opposed to small scale experimental situations. Of major interest was the ability of the CAI program to provide instruction and at the same time to record precise, extensive data on student behavior. These data were used to analyze various optimization and individualization techniques, on the level of single problems as well as over entire lessons.

The BASIC Instructional Program (BIP) was developed as a vehicle for CAI research in optimal selection of instructional material by means of an internally stored curriculum description and a model reflecting the student's changing state of knowledge and skill. BIP's design is very different from that of the AID course, specifically in its decision-making mechanisms that present material on the basis of the student's

concept-oriented history rather than in a series of ordered lessons. A student's history in the AID course consisted of a record of correct and incorrect responses to the problems in each lesson. In BIP both the student history and the curriculum organization are used in a much more dynamic way, as each is more directly related to the content of the course, described as sets of very specific programming skills.

Our work in these four areas is outlined and discussed briefly in the present report, with numerous references to detailed discussions in our technical reports and other publications.

THE IMPROVEMENT AND INDIVIDUALIZATION OF COMPUTER-ASSISTED INSTRUCTION: FINAL REPORT

Marian Beard, Avron Barr, Dexter Fletcher, and Richard C. Atkinson

The Institute for Mathematical Studies in the Social Sciences was given research support under Office of Naval Research Contract No. N00014-67-A-0012-0054 from August 1970 through July 1975, to investigate techniques in computer-assisted instruction (CAI), particularly aimed toward the optimization and individualization of instruction. Against the background of a theory of instruction, work was conducted in two subject areas: computer-controlled programs in second-language vocabulary acquisition and CAI in computer programming, allowing comparison of the quite different models of learning and optimization procedures appropriate to each.

Atkinson (1972a, 1975a) discusses the factors that need to be examined in deriving optimal instructional strategies, and identifies the key elements of a theory of instruction. The derivation of an optimal strategy requires that the instructional problem be stated in a form amenable to a decision-theoretic analysis. Analyses based on decision theory vary somewhat from field to field, but the same formal elements can be found in most of them. Stated in a general way, these elements are as follows:

1. The possible states of nature.
2. The actions that the decision maker can take to transform the state of nature.
3. The transformation of the state of nature that results from each action.

4. The cost of each action.

5. The return resulting from each state of nature.

In the context of instruction, these elements divide naturally into three groups. Elements 1 and 3 are concerned with a description of the learning process; elements 4 and 5 specify the cost-benefit dimensions of the problem; and element 2 requires that the instructional actions from which the decision maker is free to choose be precisely specified.

For the decision problems that arise in instruction, elements 1 and 3 require that a model of the learning process exist. It is usually natural to identify the states of nature with the learning states of the student. Specifying the transformation of the states of nature caused by the actions of the decision maker is tantamount to constructing a model of learning for the situation under consideration. The learning model will be probabilistic to the extent that the state of learning is imperfectly observable or the transformation or the state of learning that a given instructional action will cause is not completely predictable.

The specification of costs and returns in an instructional situation (elements 4 and 5) tends to be straightforward when examined on a short-term basis, but virtually intractable over the long term. For the short term one can assign costs and returns for the mastery of, say, certain basic reading skills, but sophisticated determinations for the long-term value of these skills to the individual and society are difficult to make. There is an important role for detailed economic analyses of the long-term impact of education, but such studies deal with issues at a more global level than we shall consider here. The present analysis

will be limited to those costs and returns directly related to a specific instructional task.

Element 2 is critical in determining the effectiveness of a decision-theory analysis; the nature of this element can be indicated by an example. Suppose we want to design a supplementary set of exercises for an initial reading program that involves both sight-word identification and phonics. Let us assume that two exercise formats have been developed, one for training on sight words, the other for phonics. Given these formats, there are many ways to design an overall program. A variety of optimization problems can be generated by fixing some features of the curriculum and leaving others to be determined in a theoretically optimal manner. For example, it may be desirable to determine how the time available for instruction should be divided between phonics and sight-word recognition, with all other features of the curriculum fixed. A more complicated question would be to determine the optimal ordering of the two types of exercises in addition to the optimal allocation of time. It would be easy to continue generating different optimization problems in this manner. The main point is that varying the set of actions from which the decision maker is free to choose changes the decision problem, even though the other elements remain the same.

Once these five elements have been specified, the next task is to derive the optimal strategy for the learning model that best describes the situation. If more than one learning model seems reasonable a priori, then competing candidates for the optimal strategy can be deduced. When these tasks have been accomplished, an experiment can be designed to determine which strategy is best. There are several possible directions

in which to proceed after the initial comparison of strategies, depending on the results of the experiment.

CRITERIA FOR A THEORY OF INSTRUCTION

Our discussion to this point can be summarized by listing four criteria that must be satisfied prior to the derivation of an optimal instructional strategy:

1. Specification of admissible instructional actions.
2. Specification of instructional objectives.
3. A measurement scale that permits costs to be assigned to each of the instructional actions and payoffs to the achievement of instructional objectives.
4. A model of the learning process.

If these four elements can be given a precise interpretation, then it is generally possible to derive an optimal instructional policy. The solution for an optimal policy is not guaranteed, but in recent years some powerful tools have been developed for discovering optimal or near optimal procedures if they exist.

The four criteria listed above, taken in conjunction with methods for deriving optimal strategies, define either a model of instruction or a theory of instruction. Whether the term theory or model is used depends on the generality of the applications that can be made. Much of the work supported by the contract has been concerned with the development of specific models for specific instructional tasks; hopefully, the collection of such models will provide the groundwork for a general theory of instruction.

In terms of the criteria listed above, it is clear that a model or theory of instruction is in fact a special case of what has come to be

known in the mathematical and engineering literature as optimal control theory or, more simply, control theory. The development of control theory has progressed at a rapid rate both in the United States and abroad, but most of the applications involve engineering or economic systems of one type or another. Precisely, the same problems are posed in the area of instruction except that the system to be controlled is the human learner, rather than a machine or group of industries. To the extent that the above four elements can be formulated explicitly, methods of control theory can be used in deriving optimal instructional strategies.

SECOND-LANGUAGE VOCABULARY ACQUISITION

Two projects involving second-language vocabulary were carried out under the contract. The first programs discussed here are based on solid mathematical theories of simple learning tasks. In particular, they attempt to optimize the memorization of translations of foreign language vocabulary items by individualizing the sequence of item presentation. A description of this instructional situation as a probabilistic Markov process is used to derive an item sequencing algorithm that facilitates significant improvement in acquisition rates. We also describe a mnemonic memorization technique that we are currently exploring in conjunction with the second-language vocabulary studies.

An Experiment on Optimal Sequencing Schemes

In this study a large set of German-English items are to be learned during an instructional session that involves a series of trials. On each trial, one of the German words is presented and the student attempts to give the English translation; the correct translation is then presented for a brief study period. A predetermined number of trials is

allocated for the instructional session, and after some intervening period a test is administered over the entire vocabulary. The problem is to specify a strategy for presenting items during the instructional session so that performance on the delayed test will be maximized.

Four strategies for sequencing the instructional material will be considered. One strategy, designated RO for random order, is to cycle through the set of items randomly; this strategy is not expected to be particularly effective, but it provides a benchmark against which to evaluate other procedures. A second strategy, designated SS for self selection, is to let the student determine for himself how best to sequence the material. In this mode, the student decides on each trial which item is to be presented.

The third and fourth schemes are based on a decision-theoretic analysis of the task. A mathematical model that provides an accurate account of vocabulary acquisition is assumed to hold in the present situation. The model is used to compute, on a trial-by-trial basis, an individual student's current state of learning. Based on these computations, items are selected for test and study so as to optimize the level of learning achieved at the termination of the instructional session. Two optimization strategies derived from this type of analysis will be examined. In one case, the computations for determining an optimal strategy are carried out assuming that all vocabulary items are of equal difficulty; this strategy is designated OE (i.e., optimal under the assumption of equal item difficulty). In the other case, the computations take into account variations in difficulty level among items;

this strategy is called OU (i.e., optimal under the assumption of unequal item difficulty). The details of these two strategies will be described later.

Both the OU and OE schemes assume that vocabulary learning can be described by a fairly simple model. We postulate that a given item is in one of three states (P, T, and U) at any moment in time. If the item is in State P, then its translation is known and this knowledge is "relatively" permanent in the sense that the learning of other items will not interfere with it. If the item is in State T, then it is also known but on a "temporary" basis; in State T the learning of other items can give rise to interference effects that cause the item to be forgotten. In State U the item is not known, and the student is unable to give a translation.

When Item i is presented the following transition matrix describes the possible change in its state:

$$L(i) = \begin{array}{c|ccc} & P & T & U \\ \hline P & 1 & 0 & 0 \\ T & x(i) & 1-x(i) & 0 \\ U & y(i) & z(i) & 1-y(i)-z(i) \end{array}$$

Rows of the matrix represent the state of the item at the start of the trial, and columns the state at the end of the trial. On a trial when some other item is presented for test and study, transitions in the state of Item i also may take place. Such transitions can occur only if the student makes an error on the other item; in that case the transition matrix applied to Item i is as follows:

$$F(i) = \begin{array}{c} \\ P \\ T \\ U \end{array} \begin{array}{c} \begin{array}{ccc} P & T & U \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1-f(i) & f(i) \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

Basically, the idea is that when some other item is presented that the student does not know, forgetting may occur for Item i if it is in State T.

Prior to conducting the experiment reported here, a pilot study was run using the same word lists and the RO procedure described above. Data from the pilot study were employed to estimate the parameters of the model; the estimates were obtained using the minimum chi-square procedures described in Atkinson (1972b). Two separate estimates of parameters were made. In one case it was assumed that the items were all equally difficult, and data from all 84 items were lumped together to obtain a single estimate of the parameter vector; this estimation procedure will be called the equal-parameter case (E case). In the second case data were separated by items, and an estimate of the parameter vector was made for each of the 84 items; this procedure will be called the unequal-parameter case (U case). The two sets of parameter estimates were then used to generate the optimization schemes previously referred to as the OE and OU procedures.

In order to formulate an "optimal" instructional strategy, it is necessary to be precise about the quantity to be maximized. For the present experiment the goal is to maximize the total number of items the student correctly translates on the delayed test. To do this, we need to specify the relationship between the state of learning at the end of the instructional session and performance on the delayed test. The assumption made here is that only those items in State P at the end

of the instructional session will be translated correctly on the delayed test; an item in State T is presumed to be forgotten during the intervening week. Thus, the problem of maximizing delayed-test performance involves maximizing the number of items in State P at the end of the instructional session.

The learning model can be used to derive equations and, in turn, compute the probabilities of being in States P, T, and U for each item at the start of any trial, conditionalized on the student's response history up to that trial. Given numerical estimates of these probabilities, a strategy for optimizing performance is to select that item for presentation that has the greatest probability of moving into State P. This strategy has been termed the "one-stage" optimization procedure because it looks ahead one trial in making decisions.

The experiment was carried out under computer control. The students participated in two sessions: an "instructional session" of approximately two hours and a briefer "delayed-test session" administered one week later. The delayed test was the same for all students and involved a test over the entire vocabulary. The instructional session was more complicated. The vocabulary items were divided into seven lists, each containing 12 German words; the lists were arranged in a round-robin order. On each trial of the instructional session a list was displayed, and the student inspected it for a brief period of time. Then one of the items on the list was selected for test and study. In the RO, OE, and OU conditions the item was selected by the computer; in the SS condition the item was chosen by the student. After an item was selected for test, the student attempted to provide a translation; then feedback

regarding the correct translation was given. The next trial began with the computer displaying the next list in the round robin, and the same procedure was repeated. The instructional session continued in this fashion for 336 trials.

The results of the experiment can be summarized as follows: Performance during the instructional session is best for the RO condition, next best for the OE condition which is slightly better than the SS condition, and poorest for the OU condition. The order of the groups is reversed on the delayed test. The OU condition is best with a correct response probability of .79; the SS condition is next with .58; the OE condition follows closely at .54; and the RO condition is poorest at .38. The observed pattern of results is what one would expect. In the SS condition, the students are trying to test themselves on items they do not know; consequently, during the instructional session, they should have a lower proportion of correct responses than students run on the RO procedure where items are tested at random. Similarly, the OE and OU conditions involve a procedure that attempts to identify and test those items that have not yet been mastered and should produce high error rates during the instructional session. The ordering of groups on the delayed test is reversed since all words are tested in a non-selective fashion; under these conditions the proportion of correct responses provides a measure of a student's mastery of the total set of vocabulary items.

The magnitude of the effects observed on the delayed test is of practical significance. The SS condition (when compared to the RO condition) leads to a relative gain of 53%, whereas the OU condition yields a relative gain of 108%. It is interesting that students were

effective in determining an optimal study sequence, but not so effective as the best of the two adaptive teaching systems.

The OU procedure is sensitive to interitem differences and consequently generates a more effective optimization strategy than the OE procedure. The OE procedure, however, is almost as effective as having the student make his own instructional decisions and far superior to a random presentation scheme.

This investigation and similar studies are reported in detail in Atkinson (1972b), Atkinson and Paulson (1972), and Paulson (1973).

Mnemonic Methods and Vocabulary Learning

When conducting vocabulary studies of the sort reported above, one is struck by the large variability in learning rates across subjects. Even Stanford students, who represent a highly selected sample from the college population, display impressively large between-subject differences. These differences may reflect differences in fundamental learning abilities, but they are also influenced by the strategies that each student brings to bear on the task. Good learners can introspect with ease about a "bag of tricks" they use in vocabulary learning whereas poor learners are unable to describe what they are doing except possibly to comment that they rehearse to themselves. The poor learners might well perform at a much higher level if they were aware of the techniques that good learners report using. With this in mind, we conducted a series of experiments on mnemonic methods for vocabulary learning. In this summary report we will only describe the nature of one of these procedures that we have called the keyword method; for a more detailed account of this research see Raugh and Atkinson (1975) and Atkinson (1975b).

The keyword method divides the study of a vocabulary item into two stages. The first stage involves associating the spoken foreign word to an English word that sounds approximately like some part of the foreign word. As an example, the Spanish word caballo (pronounced somewhat like "cob-eye-yo") contains a sound that resembles the spoken English word "eye"; we call such a similar sounding English word a keyword. The second stage involves mental imagery in which a symbolic image of the keyword interacts in a graphic way with a symbolic image of the English translation. In the case of caballo (meaning horse), one could form a mental image of something like a cyclopean eye winking in the forehead of a horse or a horse kicking a giant eye. As another example, the Spanish word for duck is pato (pronounced somewhat like "pot-o"). Employing the English word "pot" as the keyword one could imagine a duck hiding under an overturned pot with its webbed feet and tufted tail sticking out below. The method can be thought of as a chain of two links connecting a foreign word to its English translation through the mediation of a keyword. The foreign word is linked to the keyword by a similarity in sound (the acoustic link); the keyword is in turn linked to the English translation by a learner-generated mental image (the mnemonic or imagery link).

The experiments evaluating the effect of the keyword method with Spanish vocabulary items are reported in Raugh and Atkinson (1975). We have also completed a series of similar studies using a Russian vocabulary (Atkinson & Raugh, 1975). In one such experiment, two treatments were compared: subjects in the "keyword" group were supplied with an English keyword to facilitate their learning, while subjects in the control

group were given no mnemonic aid. For all subjects, the Russian item was pronounced (through a computer-controlled audio facility) as the keyword and translation, or the translation alone, were displayed on a CRT terminal screen.

Instruction sessions, lasting approximately 45 minutes per day for three days, presented a 120-word Russian vocabulary, one 40-word sub-vocabulary on each day. A daily session consisted of three study-test cycles through the 40 words. During the study phase, each Russian word was pronounced as the appropriate material was displayed (either keyword and translation or translation alone) for 10 seconds. In the test phase, the Russian word was pronounced and the program waited 10 seconds for the subject to initiate his typing of the English translation.

A comprehensive test of all 120 words was given on the fourth day, and a delayed comprehensive test was given 30 to 60 days later. The results of all the tests, during the three instruction days and for both comprehensive tests, favored the keyword condition. On each day the keyword group learned more words in two study-test trials than the control group learned in all three trials. On the comprehensive test (Day 4), the mean probability of a correct response was .72 for the keyword condition, .46 for the control. Finally, on the delayed comprehensive test, the probability correct for the keyword group was .43, for the control group, .28.

It appears that the size of the keyword effect for Russian is even larger than that for Spanish. The reason is that many subjects in the Spanish experiments had studied at least one Romance language and consequently were able to learn some of the Spanish words by using cognates

as memory aids. In Russian there are few cognates, and the keyword method appears to be even more useful. During the last year a large computer-based vocabulary drill supplement using the keyword method was offered to second-year Russian students at Stanford (see Raugh, Schupbach & Atkinson, 1975).

INSTRUCTION IN COMPUTER PROGRAMMING

This section describes two CAI courses in computer programming, both intended for college or junior college level students. The first course, which teaches the AID language, was originally developed with funds from the National Aeronautics and Space Administration; this development is described by Friend and Atkinson (1971), and Friend (1971). Continued development under ONR support used the AID course as a research vehicle for studies in optimization procedures appropriate in complex technical areas. Development of the second course, called the BASIC Instructional Program, has been supported jointly by ONR and the Advanced Research Projects Agency.

The AID Course

The course "Computer-assisted Instruction in Programming: AID" is completely self-contained and requires no supervision from a qualified instructor of programming. A brief student manual (Friend, 1972) is supplied to supplement the instruction given by computer. Teletypewriter operation is simple and can be learned from short instructions printed in the student manual.

Once the student has the teletypewriter in operation, all further instruction is given by computer under the control of a program known

as INST. This program, which is the major component of the INSTRUCT system, interprets coded lessons providing individualized, tutorial instruction to the student. This instructional system and the method of programming lessons for it are described by Friend (1971).

The AID course uses most of the features of the INSTRUCT system. The course contains 50 lessons organized into seven "lesson blocks." Each block contains five tutorial lessons, followed by a self-test and a general review. The 50th lesson is a concluding lesson independent of the blocks. The lessons vary in length from 10 to 60 exercises depending upon the content. Lessons of average length require about one hour to complete. Lesson length is completely under student control, and a student may take a few exercises or several lessons in one sitting.

One of the primary teaching strategies used in the course is the provision for student control of the sequence of instruction. Students may skip from any exercise in the course to any other exercise at any time, retracing their steps if they wish, or skipping lessons entirely. This strategy is intended to encourage the student to take responsibility for learning the concepts, not simply for progressing through a given set of exercises. Most college students are capable, and desirous, of assuming this responsibility, and the provision of student control of instruction is assumed to provide motivation.

Because of this allowance for student control, the 50 lessons may be taken in any sequence. If the student does not exercise his prerogative for choosing the sequence, the lessons are automatically sequenced for him; and it is assumed that most students will complete the lessons in the order indicated.

Besides the main strand of lessons, the course also contains review lessons, one for each of the tutorial lessons in the seven lesson blocks. These review lessons are also tutorial and cover the same concepts as do the lessons they are associated with. However, they present each concept from a slightly different viewpoint, providing additional practice in the skills to be learned. In general, each lesson covers five or six related concepts. In review lessons, the student may review whichever concepts he wishes, in any order he chooses. In fact, he must choose the order; there is no automatic sequencing provided by the program. At the end of each tutorial lesson, the student is asked if he wants to review any of the ideas covered in the lesson just completed. The student need not wait for these reminders, of course, since he can call for any review, or any exercise in any review, whenever he wishes.

Also associated with each tutorial lesson is a summary of the lesson, and the student is reminded at the end of each lesson that summaries are available at his option. In addition to the main strand of lessons, the reviews, and the summaries, there is a strand of "extra-credit" problems containing more difficult programming problems to be solved by the more capable students.

The inclusion of review lessons is a gross method for providing individualized remediation. A more sensitive means of individualizing remediation is used within the lessons themselves, where non-optional remedial sequences of exercises are given automatically to students who demonstrate an inadequate understanding of the material being taught. Because of this automatic remediation, different students may receive different numbers of exercises in a given lesson.

A student who makes an incorrect response to an exercise may not need an entire sequence of remedial exercises. He may profit from a single specific corrective message, pointing out the error and allowing him another try at the same problem. This kind of specific correction is used for most exercises in the course. Messages are provided, not for all possible incorrect responses, but for those incorrect responses judged to be most likely to occur.

The curriculum and student control features are described more fully by Friend, Fletcher, and Atkinson (1972) and Friend (1973a,b). Extensive analysis of students' problem-solving behavior, focussing on problem difficulty and diversity of student solutions to programming problems, is presented in Friend (1975).

An important aspect of the research in individualization involved mechanisms allowing students to exercise a considerable degree of control over the content and sequencing of instructional material, as mentioned above. A study described by Beard, Lorton, Searle, and Atkinson (1973) was conducted to compare a student-selection scheme against two strategies of computer control. One major finding of this study was that students do not choose to exercise much control over the material presented to them; those students who were allowed their choice of lessons consistently followed the path of the ordered lessons. More significant, in light of the direction taken since that time, was the conclusion that the AID course itself was not ideally suited to investigation of sophisticated individualization schemes. First, since the curriculum is clearly laid out in a pedagogically sound, linear order, it actually discourages students from making different choices. Second, the instructional program

is not directly linked to the AID interpreter, through which the students write their own programs, and thus the course cannot provide assistance or instruction during the problem-solving activity itself. As the student writes his program, his only sources of assistance are the error messages provided by the non-instructional AID interpreter.

An inadequacy of the AID course, especially for research purposes, is its limited ability to characterize individual students' knowledge of specific skills, and its inability to relate students' skills to the curriculum as anything more than a ratio of problems correct to problems attempted. The program cannot make fine distinctions between a student's strengths and weaknesses, and cannot present instructional material specifically appropriate to that student beyond "harder" or "easier" lessons. In order to explore the effects of different curriculum selection strategies in more detail, we developed a new introductory course in computer programming capable of representing both its subject matter and student performance more adequately.

The BASIC Instructional Program is a stand-alone, fully self-contained course in BASIC programming at the high school/college level developed over the past two years with the assistance of over 300 undergraduates who have taken the course at DeAnza College, the University of San Francisco, and Stanford. Our classroom experiences developing BIP as an instructional vehicle were described by Barr, Beard, Lorton, and Atkinson (1974a,b). BIP's major features are:

- A monitored BASIC interpreter, written by the project staff, which allows the instructional system maximal knowledge about student errors.

- A HINT system that gives both graphic and textual aid in problem solving.

- Individualized task selection based on a Curriculum Information Network, which describes the problems in terms of fundamental skills. Problems are selected using a model of the student's acquisition of the skills required by his earlier programming problems.

- A curriculum consisting of approximately 100 well-written, interesting programming problems at widely varying levels of difficulty.

The tutorial programming laboratory environment supported by BIP is described fully by Barr, Beard, and Atkinson (1975a). To the student seated at a terminal, BIP looks very much like a typical timesharing BASIC operating system. The BASIC interpreter, written especially for BIP, analyzes each program line after the student types it, and notifies the student of syntax errors. When the student runs his program, it is checked for structural illegalities, and then, during runtime, "execution" errors are indicated. A file storage system, a calculator, and utility commands are available.

BIP's Curriculum Information Network

In much of the current research in tutorial CAI, generative CAI, and mixed-initiative natural language dialogues, the central problem is the "representation" of the subject domain, which is also a fundamental concern of research in cognitive psychology and artificial intelligence. The goal is to provide a representation of the subject matter that is sufficient for individualized tutoring and also has a realistic and manageable computer implementation. In technical subjects, development of skills requires the integration of facts, not just their memorization, and the organization of instructional material is crucial for effective instruction in these areas.

The Curriculum Information Network (CIN) is intended to provide the instructional program with an explicit representation of the structure of an author-written curriculum (Barr, Beard, & Atkinson, 1975c). It contains the interrelations between the problems which the author would have used implicitly in determining his "branching" schemes. It allows meaningful modelling of the student's progress along the lines of his developing skills, not just his history of right and wrong responses, without sacrificing the motivational advantages of human organization of the curriculum material. For example, in the BIP course, the CIN consists of a complete description of each of the 100 programming problems in terms of the skills developed in solving the problems. Thus, the instructional program can monitor the student's progress on these skills and choose the next task with an appropriate group of new skills. An intermediate step is introduced between recording the student's history and selecting his next problem: the network becomes a model of the student's state of knowledge, since it has an estimate of his ability in the relevant skills, not just his performance on the problems he has completed. Branching decisions are based on this model instead of being determined simply by the student's success/failure history on the problems he has completed.

In this way, a problem can be presented for different purposes to students with different histories. The flexibility of the curriculum is of course multiplied as a result. More importantly, the individual problems in the curriculum can be more natural and meaningful; they do not necessarily involve only one skill or technique. In frame-type curriculums this one-dimensionality of the problems has a constricting

effect. In essence, the network as implemented in BIP is a method of describing a "real" curriculum in terms of the specific skills that can be identified as a student's problem areas.

BIP's Instructional Environment

Computer programming, like most other technical subjects, is better learned through experience than through direct instruction, especially if that experience can be paced at a speed suited to the individual student. Throughout the BIP course, the primary emphasis is placed on the solution of problems presented in the tasks. BIP does not present a sequence of instructional statements followed by questions. Instead, a problem is described and the student is expected to write his own BASIC program to solve it. As he develops his BASIC program for each task, the student is directed to appropriate sections of the student manual (Beard & Barr, 1974) for full explanations of BASIC statements, programming structures, etc. He is also encouraged to use the numerous student-oriented features available, such as an interactive debugging facility and various "help" options described in Barr, Beard, and Atkinson (1975b).

When a student enters the course he finds himself in task "GREENFLAG", which requires a two-line program solution. The problem, as he is told, is worked out in great detail in the BIP student manual. Thus, the trauma of being told to "write a program that..." in his first session is alleviated by following the model dialogue, in which many typical mistakes are illustrated, yet his hands-on programming experience begins immediately. When he has finished the task by successfully running his program, the student proceeds by requesting "MORE". His progress is

evaluated after each task. In the "Post Task Interview" he is asked to indicate whether or not he needs more work on the skills required by the task, which are listed separately for him.

As soon as the student completes GREENFLAG, therefore, the instructional program knows something about his own estimation of his abilities. In addition, for all future tasks his solution is evaluated (by means of comparing its output with that of the model solution run on the same test data) and the results are stored with each skill required by the task. The program then has two measures of the student's progress in each skill--his self-evaluation and its own comparison-test results.

A student progresses through the curriculum by writing, and running, a program that solves the problem presented on his terminal. Virtually no limitations are imposed on the amount of time he spends, the number of lines he writes, the number of errors he is allowed to make, the number of times he chooses to execute the program, the changes he makes within it, etc. The task on which he is working is stored on a stack-like structure, so that he may work on another task, for whatever reason, and return to the previous task automatically. The curriculum structure can accommodate a wide variety of student aptitudes and skills. Most of the curriculum-related options are designed with the less competent student in mind. A more independent student may simply ignore the options. Thus, BIP gives students the opportunity to determine their own "challenge levels" by making assistance available but not inevitable.

BIP offers the student considerable flexibility in making his own task-related decisions. He may ask for hints and subtasks to help him get started in solving the given problem, or he may ponder the problem

on his own, using only the manual for additional information. He may request a different task by name, in the event that he wishes to work on it immediately, either completing the new task or not, as he chooses. On his return, BIP tells him the name of the again-current task, and allows him to have its text printed to remind him of the problem he is to solve. The student may request the model solution for any task at any time, but BIP will not print the model for the current task unless the student has exhausted the available hints and subtasks. Taken together, the curriculum options allow for a wide range of student preferences and behaviors.

The BIP program has been running successfully with both junior college and university students. However, the program is still very much in an experimental stage. From a psychological viewpoint, the principal research issues deal with (1) procedures for obtaining on-line estimates of student abilities as represented in the information network, and (2) alternative methods for using the current estimates in the information network to make instructional decisions. For a more complete description of our recent work on BIP and a review of our plans for continued research see Barr, Beard, and Atkinson (1975d).

CONCLUDING REMARKS

The projects described in this paper have one theme in common, namely, developing computer-controlled procedures for optimizing the instructional process. For several of the instructional tasks considered here, mathematical models of the learning process were formulated which made it possible to use formal methods in deriving optimal policies.

In other cases the "optimal schemes" were not optimal in a well-defined sense, but were based on our intuitions about learning and some relevant experiments. In a sense, the diversity represented in these examples corresponds to the state of the art in the field of instructional design. For some tasks we can use psychological theory to help define optimal procedures; for others our intuitions, modified by experiments, must guide the effort. Hopefully, our understanding of these matters will increase as more projects are undertaken to develop sophisticated instructional programs.

Some have argued that any attempt to devise optimal strategies is doomed to failure, and that the learner himself is the best judge of appropriate instructional actions. We are not sympathetic to a learner-controlled approach to instruction, because we believe its advocates are trying to avoid the difficult but challenging task of developing a viable theory of instruction. There obviously is a place for the learner's judgments in making instructional decisions; for example, such judgments play an important role in several parts of our BTP course. However, using the learner's judgment as one of several items of information in making instructional decisions is different from proposing that the learner should have complete control. Results presented in this paper and those cited in Beard et al. (1973) indicate that the learner is not a particularly effective decision maker in guiding the learning process.

At the beginning of this report we defined the four criteria that must be satisfied before an optimal instructional procedure can be derived using formal methods. For the types of instructional situations dealt with during the life of this contract specifications can be offered

for the first three elements. However, the fourth element--the specification of a model of the learning process--represents a major obstacle. Our theoretical understanding of learning is so limited that only in very special cases can a model be specified in enough detail to enable the derivation of optimal procedures. Until we have a much deeper understanding of the learning process, the identification of truly effective strategies will not be possible.

However, an all-inclusive theory of learning is not a prerequisite for the development of optimal procedures. What is needed is a model that captures the essential features of that part of the learning process being tapped by a given instructional task. Even models that have been rejected on the basis of laboratory investigations may be useful in deriving instructional strategies. Several of the learning models considered in this paper have proven unsatisfactory when tested in the laboratory and evaluated using standard goodness-of-fit criteria; nevertheless, the optimal strategies they generate are often quite effective. Our own preference is to formulate as complete a learning model as intuition and data will permit and then use that model to investigate optimal procedures. When possible the learning model should be represented in the form of mathematical equations, but otherwise as a set of statements in a computer-simulation program. The main point is that the development of a theory of instruction cannot progress if one holds the view that a comprehensive theory of learning is a prerequisite. Rather, advances in learning theory will affect the development of a theory of instruction, and conversely the development of a theory of instruction will influence the direction of research on learning.

REFERENCES

Atkinson, R. C. Ingredients for a theory of instruction. American Psychologist, 1972a, 27, 921-931.

Atkinson, R. C. Optimizing the learning of a second-language vocabulary. Journal of Experimental Psychology, 1972b, 96, 124-129.

Atkinson, R. C. Adaptive instructional systems: Some attempts to optimize the learning process. In D. Klahr (Ed.), Cognition and instruction. Hillsdale, New Jersey: Erlbaum Associates, 1975a, in press.

Atkinson, R. C. Mnemotechnics in second-language learning. American Psychologist, 1975b, in press.

Atkinson, R. C., & Paulson, J. A. An approach to the psychology of instruction. Psychological Bulletin, 1972, 78, 49-61.

Atkinson, R. C., & Raugh, M. R. An application of the mnemonic keyword method to the acquisition of a Russian vocabulary. Journal of Experimental Psychology, 1975, 104, 126-133.

Barr, A., & Atkinson, R. C. Adaptive instructional strategies. In H. Spada and W. R. Kempf (Eds.), Structural models of thinking and learning: Proceedings of the IPN-Symposium 7. Bern, West Germany: Hans Huber Publishers, 1976, in press.

Barr, A., Beard, M., & Atkinson, R. C. A rationale and description of a CAI program to teach the BASIC programming language. Instructional Science, 1975a, 4, 1-31.

Barr, A., Beard, M., & Atkinson, R. C. Computer-assisted instruction in BASIC programming. Proceedings of the ACM Computer Science Conference, Washington, D.C., 1975b.

Barr, A., Beard, M., & Atkinson, R. C. Information networks for CAI curriculums. Proceedings of the Second IFIP World Conference on Computers in Education, Marseilles, France, 1975c.

Barr, A., Beard, M., & Atkinson, R. C. The computer as a tutorial laboratory: The Stanford BIP project (Tech. Rep. No. 260). Stanford, Calif.: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1975d.

Barr, A., Beard, M., Lorton, P., & Atkinson, R. C. Tutorial CAT in technical areas: Skill development using an information network. Proceedings of the Fifth Annual Conference on Computers in the Undergraduate Curriculum, Pullman, Wash., 1974a.

Barr, A., Beard, M., Lorton, P., & Atkinson, R. C. A college level course in BASIC. In G. Goos and J. Hartmania (Eds.), Lecture notes on computer science: RGU '74 (Vol. 17). Berlin, West Germany: Springer-Verlag, 1974b.

Beard, M., & Barr, A. BIP student manual. Institute for Mathematical Studies in the Social Sciences, Stanford University, 1974.

Beard, M., Lorton, P., Searle, B. W., & Atkinson, R. C. Comparison of student performance and attitude under three lesson-selection strategies in computer-assisted instruction (Tech. Rep. No. 222). Stanford, Calif.: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1973.

Friend, J. INSTRUCT coders' manual (Tech. Rep. No. 172). Stanford, Calif.: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1971.

Friend, J. AID student manual. Institute for Mathematical Studies in the Social Sciences, Stanford University, 1972.

Friend, J. Computer-assisted instruction in programming: A curriculum description (Tech. Rep. No. 211). Stanford, Calif.: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1973a.

Friend, J. 100 programming problems. Institute for Mathematical Studies in the Social Sciences, Stanford University, 1973b.

Friend, J. Programs students write (Tech. Rep. No. 257). Stanford, Calif.: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1975.

Friend, J., & Atkinson, R. C. CAI in programming: AID (Tech. Rep. No. 184). Stanford, Calif.: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1971.

Friend, J., Fletcher, J. D., & Atkinson, R. C. Student performance in computer-assisted instruction in programming (Tech. Rep. No. 184). Stanford, Calif.: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1972.

Paulson, J. A. An evaluation of instructional strategies in a simple learning situation (Tech. Rep. No. 209). Stanford, Calif.: Institute for Mathematical Studies in the Social Sciences, Stanford, University, 1973.

Raugh, M. R., & Atkinson, R. C. A mnemonic method for learning a second-language vocabulary. Journal of Educational Psychology, 1975, 67, 1-16.

Raugh, M. R., Schupbach, R. D., & Atkinson, R. C. Teaching a large Russian language vocabulary by the mnemonic keyword method (Tech. Rep. No. 256). Stanford, Calif.: Institute for Mathematical Studies in the Social Sciences, Stanford University, 1975.

TECHNICAL REPORTS

ONR Contract No. N00014-67-A-0012-0054

Institute for Mathematical Studies in the Social Sciences

184 Friend, J., Fletcher, J. D., & Atkinson, R. C. Student performance in computer-assisted instruction in programming

187 Atkinson, R. C. Ingredients for a theory of instruction

209 Paulson, J. A. An evaluation of instructional strategies in a simple learning situation

211 Friend, J. Computer-assisted instruction in programming: A curriculum description

222 Beard, M., Lorton, P., Searle, B. W., & Atkinson, R. C. Comparison of student performance and attitude under three lesson-selection strategies in computer-assisted instruction

224 Raugh, M. R., & Atkinson, R. C. A mnemonic method for the acquisition of a second-language vocabulary

228 Barr, A., Beard, M., & Atkinson, R. C. A rationale and description of the BASIC instructional program

237 Atkinson, R. C., & Raugh, M. R. An application of the mnemonic keyword method to the acquisition of a Russian vocabulary

256 Raugh, M. R., Schupbach, R. D., & Atkinson, R. C. Teaching a large Russian language vocabulary by the mnemonic keyword method

257 Friend, J. Programs students write

260 Barr, A., Beard, M., & Atkinson, R. C. The computer as a tutorial laboratory: The Stanford BIP project

DISTRIBUTION LIST

Navy

4 Dr. Marshall J. Farr, Director
Personnel & Training Research Programs
Office of Naval Research (Code 458)
Arlington, VA 22217

1 ONR Branch Office
495 Summer Street
Boston, MA 02210
Attn: Research Psychologist

1 ONR Branch Office
1030 East Green Street
Pasadena, CA 91101
Attn: E. E. Gloye

1 ONR Branch Office
536 South Clark Street
Chicago, IL 60605

6 Director
Naval Research Laboratory
Code 2627
Washington, DC 20390

12 Defense Documentation Center
Cameron Station, Building 5
5010 Duke Street
Alexandria, VA 22314

1 LCDR Charles J. Theisen, Jr. MSC, USN
4024
Naval Air Development Center
Warminster, PA 18974

1 Chief of Naval Reserve
Code 3055
New Orleans, LA 70146

1 Navy Personnel Research and
Development Center
San Diego, CA 92152
Attn: Dr. J. D. Fletcher

1 Dr. Lee Miller
Naval Air Systems Command
AIR-413E
Washington, DC 20361

1 Chief
Bureau of Medicine & Surgery
Research Division (Code 713)
Washington, DC 20372

1 Chairman
Behavioral Science Department
Naval Command & Management
Division
U.S. Naval Academy
Luce Hall
Annapolis, MD 21402

1 Chief of Naval Education &
Training
Naval Air Station
Pensacola, FL 32508
Attn: Capt. Bruce Stone, USN

1 Mr. Arnold Rubinstein
Naval Material Command (NAVMAT
03424)
Room 820, Crystal Plaza No. 6
Washington, DC 20360

1 Chief of Naval Operations (OP-987E)
Department of the Navy
Washington, DC 20350

1 Superintendent
Naval Postgraduate School
Monterey, CA 93940
Attn: Library (Code 2124)

1 Chief of Naval Technical Training
Naval Air Station Memphis (75)
Millington, TN 38054
Attn: Dr. Norman J. Kerr

1 Commanding Officer
Service School Command
U.S. Naval Training Center
San Diego, CA 92133
Attn: Code 3030

1 Dr. William L. Maloy
Principal Civilian Advisor
for Education & Training
Naval Training Command, Code 01A
Pensacola, FL 32508

1 Dr. Alfred F. Smode, Staff Consultant
Training Analysis & Evaluation Group
Naval Training Equipment Center
Code N-00T
Orlando, FL 32813

1 Chief of Naval Training Support
Code N-21
Building 45
Naval Air Station
Pensacola, FL 32508

1 Dr. Robert French
Naval Undersea Center
San Diego, CA 92132

1 LCDR C. F. Logan, USN
F-14 Management System
COMFITAEWWINGPAC
Nas Miramar, CA 92145

1 Dr. John Ford
Navy Personnel R&D Center
San Diego, CA 92152

5 Navy Personnel R&D Center
San Diego, CA 92152
Attn: Code 10

1 Capt. D. M. Gragg, MC, USN
Head, Educational Programs Development
Department
Naval Health Sciences Education and
Training Command
Bethesda, MD 20014

Army

1 Armed Forces Staff College
Norfolk, VA 23511
Attn: Library

1 Commandant
United States Army Infantry School
Fort Benning, GA 31905
Attn: ATSH-DET

1 Mr. James Baker
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

1 Dr. Ralph Dusek
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

1 Dr. Leon H. Nawrocki
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

1 Dr. J. E. Uhlaner, Technical
Director
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

1 Dr. Joseph Ward
U.S. Army Research Institute
1300 Wilson Boulevard
Arlington, VA 22209

1 HQ USAREUR & 7th Army
OICSOPS
USAREUR Director of GED
APO New York 09403

Air Force

1 Dr. G. A. Eckstrand (AFHRL/AS)
Wright-Patterson AFB
Ohio 45433

1 Dr. Ross L. Morgan (AFHRL/AST)
Wright-Patterson AFB
Ohio 45433

1 Dr. Martin Rockway (AFHRL/TT)
Lowry AFB
Colorado 80230

1 AFOSR/NL
1400 Wilson Boulevard
Arlington, VA 22209

1 Commandant
USAF School of Aerospace Medicine
Aeromedical Library (SUL-4)
Brooks AFB, TX 78235

1 Dr. Sylvia R. Mayer (MCIT)
Headquarters Electronic Systems Division
LG Hanscom Field
Bedford, MA 01730

1 Capt. Jack Thorpe, USAF
Flying Training Division (HRL)
Williams AFB, AZ 85224

Marine Corps

1 Mr. E. A. Dover
Manpower Measurement Unit (Code MPI)
Arlington Annex, Room 2413
Arlington, VA 20380

1 Commandant of the Marine Corps
Headquarters, U.S. Marine Corps
Code MPI-20
Washington, DC 20380

1 Director, Office of Manpower Utilization
Headquarters, Marine Corps (Code MPU)
MCB (Building 2009)
Quantico, VA 22134

1 Dr. A. L. Slafkosky
Scientific Advisor (Code RD-1)
Headquarters, U.S. Marine Corps
Washington, DC 20380

1 Chief, Academic Department
Education Center
Marine Corps Development and Education Command
Marine Corps Base
Quantico, VA 22134

Other DOD

1 Lt. Col. Henry L. Taylor, USAF
Military Assistant for Human Resources
OAD (E&LS) ODDR&E
Pentagon, Room 3D129
Washington, DC 20301

1 Dr. Harold F. O'Neil, Jr.
Advanced Research Projects Agency
Human Resources Research Office
1400 Wilson Boulevard, Room 625
Arlington, VA 22209

1 Helga L. Yeich
Advanced Research Projects Agency
Manpower Management Office
1400 Wilson Boulevard
Arlington, VA 22209

Other Government

1 Dr. Erik McWilliams, Program Manager
Technology and Systems, TIE
National Science Foundation
Washington, DC 20550

1 Dr. Andrew R. Molnar
Technological Innovations in Education
National Science Foundation
Washington, DC 20550

1 Dr. Marshall S. Smith
Asst. Acting Director
Program on Essential Skills
National Institute of Education
Brown Bldg., Room 815
19th and M St., N.W.
Washington, DC 20208

Miscellaneous

1 Dr. Scarvia B. Anderson
Educational Testing Service
17 Executive Park Drive, N.E.
Atlanta, GA 30329

1 Dr. John Annett
The Open University
Milton Keynes
Buckinghamshire, England

1 Dr. Richard Snow
School of Education
Stanford University
Stanford, CA 94305

1 Dr. Norman Cliff
Department of Psychology
University of Southern California
University Park
Los Angeles, CA 90007

1 Dr. Ronald P. Carver
School of Education
University of Missouri
Kansas City, MO 64110

1 ERIC
Processing and Reference Facility
4833 Rugby Avenue
Bethesda, MD 20014

1 Dr. Victor Fields
Department of Psychology
Montgomery College
Rockville, MD 20850

1 Dr. A. Glickman
American Institutes for Research
Foxhall Square
3301 New Mexico Avenue, N.W.
Washington, DC 20016

1 Dr. Ruth Day
Department of Psychology
Yale University
New Haven, CT 06520

1 Dr. Robert Glaser, Director
Learning Research & Development
Center
University of Pittsburgh
Pittsburgh, PA 15213

1 Dr. Robert Vineberg
HumRRO Western Division
27857 Berwick Drive
Carmel, CA 93921

1 LCol. C. R. J. Lafleur, Director
Personnel Applied Research
National Defence HQ
Ottawa, Canada K1A 0K2

1 Dr. Henry J. Hamburger
School of Social Sciences
University of California
Irvine, CA 92664

1 HumRRO
Division No. 3
P.O. Box 5787
Presidio of Monterey, CA 93940

1 Dr. Lawrence B. Johnson
Lawrence Johnson & Associates, Inc.
200 S Street, N.W., Suite 502
Washington, DC 20009

1 Dr. Steven W. Keele
Department of Psychology
University of Oregon
Eugene, OR 97403

1 Dr. David Klahr
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

1 Dr. Alma E. Lantz
University of Denver
Denver Research Institute
Industrial Economics Division
Denver, CO 80210

1 Dr. Robert R. Mackie
Human Factors Research, Inc.
6780 Cortona Drive
Santa Barbara Research Park
Goleta, CA 93017

1 Dr. Donald A. Norman
Center for Human Information Processing
University of California, San Diego
La Jolla, CA 92037

1 Mr. Brian McNally
Educational Testing Service
Princeton, NJ 08540

1 Mr. A. J. Pesch, President
Eclectech Associates, Inc.
P.O. Box 178
North Stonington, CT 06359

1 Dr. Joseph W. Rigney
Behavioral Technology Laboratories
University of Southern California
3717 South Grand
Los Angeles, CA 90007

1 Dr. Leonard L. Rosenbaum, Chairman
Department of Psychology
Montgomery College
Rockville, MD 20850

1 Dr. George E. Rowland
Rowland & Company, Inc.
P.O. Box 61
Haddonfield, NJ 08033

1 Dr. Arthur I. Siegel
Applied Psychological Services
404 East Lancaster Avenue
Wayne, PA 19087

1 Mr. Dennis J. Sullivan
725 Benson Way
Thousand Oaks, CA 91360

1 Dr. Benton J. Underwood
Department of Psychology
Northwestern University
Evanston, IL 60201

1 Dr. David J. Weiss
Department of Psychology
University of Minnesota
Minneapolis, MN 55455

1 Dr. Arnold F. Kanarick
Honeywell, Inc.
2600 Ridge Parkway
Minneapolis, MN 55413

1 Dr. Carl R. Vest
Battelle Memorial Institute
Washington Operations
2030 M Street, NW
Washington, DC 20036

1 Dr. Roger A. Kaufman
United States International
University
Graduate School of Leadership and
Human Behavior
Elliott Campus
San Diego, CA 92124

1 Dr. Allan M. Collins
Bolt Beranek and Newman, Inc.
50 Moulton Street
Cambridge, MA 02138