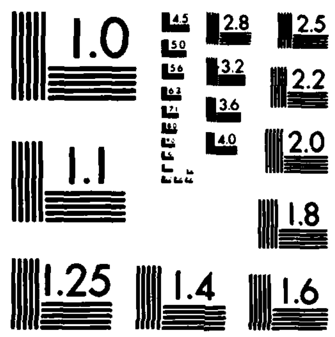
AD-A172 021 MCATL: A LANGUAGE FOR AUTHORING COMPUTERIZED ADAPTIVE TESTS(U) ASSESSMENT SYSTEMS CORP ST PAUL MN C D VALE 11 NOV 85 TR-85-3-ONR N00014-83-C-0634 1/1

UNCLASSIFIED F/G 9/2 NL


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

[illegible]

unclassified
SECURITY CLASSIFICATION OF THIS PAGE

AD-A172 021

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) TR ~~ONR~~-85-3	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Assessment Systems Corporation 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research
6c. ADDRESS (City, State, and ZIP Code) 2233 University Avenue, Suite 310 St. Paul, MN 55114	7b. ADDRESS (City, State, and ZIP Code) 800 N. Quincy Street, Code 442 Arlington, VA 22217
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Personnel and Training Research 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-83-C-0634
8c. ADDRESS (City, State, and ZIP Code) Office of Naval Research 800 N. Quincy Street, Code 442 Arlington, VA 22217	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. — PROJECT NO. N 507-002 — TASK NO. — WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
MCATL: A Language for Authoring Computerized Adaptive Tests

12. PERSONAL AUTHOR(S)
C. David Vale

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
technical report	FROM 9/1/83 TO 11/11/85	85 Nov 11	15

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	adaptive testing, authoring language, computerized testing, IRT, item response theory, MicroCAT, MCATL, tailored testing, test development, testing language, testing software

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

The specification of a computerized adaptive test, like the specification of computer-assisted instruction, is easier and can be done by personnel who are not proficient in computer programming if an authoring language is provided. The Minnesota Computerized Adaptive Testing Language (MCATL) is an authoring language specifically designed for specifying adaptive tests. Its fourteen statements can be grouped into five functions: test division, administration control, scoring, reporting, and customizing. The first four categories provide statements for specifying most types of adaptive tests with minimal programming effort; the fifth category provides an interface with standard programming languages for tests that cannot be directly specified in MCATL. A formal specification of MCATL in Backus Naur Form and a practical example are provided.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT [✓ UNCLASSIFIED/UNLIMITED] ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Charles E. Davis	22b. TELEPHONE (Include Area Code) 202-696-4046 — 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
unclassified

ABSTRACT

The specification of a computerized adaptive test, like the specification of computer-assisted instruction, is easier and can be done by personnel who are not proficient in computer programming if an authoring language is provided. The Minnesota Computerized Adaptive Testing Language (MCATL) is an authoring language specifically designed for specifying adaptive tests. Its fourteen statements can be grouped into five functions: test division, administration control, scoring, reporting, and customizing. The first four categories provide statements for specifying most types of adaptive tests with minimal programming effort; the fifth category provides an interface with standard programming languages for tests that cannot be directly specified in MCATL. A formal specification of MCATL in Backus Naur Form and a practical example of MCATL are provided.

TABLE OF CONTENTS

INTRODUCTION

Computerized adaptive testing (CAT; Weiss, 1978, 1980, 1985) is a form of psychological testing in which the computer chooses the items that are most appropriate for each examinee. Like more conventional paper-and-pencil tests, computerized tests must be constructed or authored. Paper-and-pencil authoring requires item selection, formatting, and typing. An author may choose item cards from an item bank, sort them into the desired order, and give them to a typist to produce the final test. Authoring a computerized test can be considerably more complex. In the early days of computerized testing, tests were authored in programming languages designed for general-purpose computation. Items were coded as print statements in programs. Although this process resulted in satisfactory tests, it required long programs and the services of a computer programmer.

Computer-assisted instruction (CAI) developed simultaneously with CAT. Since the underlying theory and processes associated with CAI were considerably simpler than those of CAT, CAI became commercially viable much earlier. From the point of view of authoring and administration, CAT and CAI have some similarities. At the most basic level, presentation of a CAI instructional screen and acceptance of a examinee's response is no different from presentation of a test item and acceptance of an examinee's response. Like CAT tests, CAI instruction must be authored on a computer before it can be administered to examinees. Because the use of CAI was so widespread and its content so voluminous, it was not feasible for all instruction to be authored by programmers. Nor was it desirable from the instructional author's perspective because a certain amount of interactive control over the material is lost when it is submitted to a programmer. To make it possible for a non-programmer to enter instruction, a new type of programming language, typically referred to as an authoring language, was developed.

In essence, all authoring languages attempt to provide the instructional author with a set of utilities specifically designed for creating instruction. A CAI authoring language at its most basic level has to provide a means for presenting instructional material on a computer display, accepting a response from the examinee, assigning his or her response to one of a finite number of categories, and mechanically branching as a function of the response given. These functions are also required of a CAT language.

In spite of the similarities between CAI and CAT, a standard CAI authoring system is ill suited to the development of adaptive tests because it has many features that are inappropriate for CAT and it lacks several features that are necessary for CAT. An authoring system developed for CAI must provide substantial flexibility in the item formats that it can present and considerable latitude in the types of responses it can accept. Motion or animation in the instructional display is common for CAI. Similarly, a free-response format is common for responding to CAI. Branching and instruction selection is very rudimentary, however; the branching algorithms for CAI are rarely more complicated than a simple mechanical branch to a new instructional item based on the response to the previous one. Furthermore, instructional sequences are usually not scored, and if they are, scoring is very rudimentary.

In the current state of the art, adaptive tests rarely require complex visual displays and are typically limited to a multiple-choice response format. This is due not so much to the psychometrics involved, but rather to the long tradition of objective testing in the paper-and-pencil medium, in which limited formats were available. It will probably be several years before psychological tests are developed to effectively exploit the computer's capabilities for more complex item types.

Item selection and scoring for CAT are considerably more complex. Because each examinee receives a different set of test items, the conventional proportion-correct score is inadequate. In effect, a good adaptive test attempts to choose items for a particular examinee for which he or she has approximately a 50% chance of knowing the correct answer. If the testing procedure is effective, there will be virtually no variance in proportion-correct scores among examinees. Scoring is thus typically built upon mathematically complex and computationally tedious statistical models. Depending on the purpose of the test, item selection may be as simple as the mechanical branching of a CAI program, but it usually involves selecting items to optimize a computationally tedious statistical function.

The MicroCAT^{tm} Testing System is one of a small number of CAT systems available for complete general-purpose test development. It may be the only CAT system available that contains a complete test authoring system for off-the-shelf microcomputer hardware. MCATL, the *Minnesota Computerized* Adaptive Testing Language, is the authoring language used by the MicroCAT system to specify adaptive tests. This report describes MCATL and the process by which it was developed.

LANGUAGE DESIGN

The design of any programming environment involves trade-offs between the investment in developing the system itself, the ease of program development within the system, and the efficiency with which programs developed in the system will execute. For minimal investment in system development and maximum efficiency of execution, machine language is ideal. For ease of use, high-level metaphorical programming environments are best. An example of a metaphorical programming environment is one in which the computer screen is made to resemble a desk top. Within the metaphor, the user can accomplish tasks such as shuffling papers on his or her desk with relative ease. The problem with such high-level environments is that they are usually limited to functions within their metaphors (in this case, what can be done on the top of a desk), they are very expensive to develop, and they require considerable computer power to execute. Between the two extremes of machine language and metaphorical programming environments are high-level programming languages and various kinds of authoring languages like MCATL.

The primary objective in the design of MCATL was to provide a reasonably friendly environment for authoring tests. It had to provide a framework of functions for performing authoring tasks. The design of MCATL thus began with an analysis of the authoring process, which, in a computerized mode, amounts to structuring the presentation of test items and specifying how responses to the items are to be processed.

Test administration can be characterized by three functions: presentation, scoring, and reporting. Presentation includes item selection and test termination. The criteria by which items should be selected or the rules for mechanically branching from item to item or subtest to subtest must be determined. Additionally, a test termination rule (a logical decision rule) to be evaluated after each item is administered must be chosen to determine when testing should stop.

Vale (1981) suggested that the majority of CAT strategies could be grouped into a simple structural taxonomy with three main categories: inter-item branching, inter-subtest branching, and model-based branching. In a test based on inter-item branching, items are selected by branching from the last item administered to one of two new items based solely on whether the response to the preceding item was correct or incorrect. In a test based on inter-subtest branching, the branching is from one set of items to another set of items based on a score from the previous set of items. Vale further subdivided this category into reentrant and non-reentrant branching. In a non-reentrant branching strategy, an entire subtest is administered before branching occurs. In a reentrant strategy, only one item within a subtest is administered when that subtest is branched to. Each time a subtest is branched to, the next sequential unadministered item is administered. The final category, model-based branching, selects items by searching the entire item pool to determine which item optimizes the statistical criterion function.

MCATL evolved from TCL (Test Control Language), an early prototype language for specifying adaptive tests (Vale, 1981). MCATL was designed to more closely follow the structure of Vale's taxonomy and to eliminate several shortcomings of TCL. First, because TCL did not have a subtest structure, it was difficult to administer several independent tests within a testing session. A second test could be included only by explicitly resetting all of the pointers and the scores. Second, inter-subtest branching was unstructured and could be accomplished only with conditional branches to labels in the test specification. This was especially clumsy for reentrant inter-subtest branching in which pointer variables had to be explicitly set and re-set by the test author. Although the process worked, the resulting specification was difficult to read and required a fair amount of programming skill. Third, there was no separation of scoring functions from score variables. For example, a test could not use separate Bayesian priors for item selection and scoring. Fourth, TCL had very limited reporting capabilities; the report produced was a fixed-format numeric listing of scores. Furthermore, it was difficult, if not impossible, to get scores printed after each item was administered rather than at the end of the test. Fifth, TCL had no text capabilities available for interpreting the test scores for the examinee. Finally, TCL offered no opportunity for customization; strategies that could not be specified in TCL could not be developed within the TCL system.

LANGUAGE DESCRIPTION

MCATL is a line-oriented language. All 80 characters of a line are used, and an attempt to read the 81st character is an end of line. If an ampersand appears in a line, the remaining characters are ignored, and the next line is scanned as a continuation of the previous line. The exclamation point has the same effect as an end of line, and the remaining characters are ignored.

Figure 1 shows the formal syntax of MCATL in Backus Naur Form (BNF; cf., Jensen & Wirth, 1974). The functions of the statements and their uses are described in detail in the *User's Manual for the MicroCAT Testing System* (Assessment Systems Corporation, 1984). The statements are briefly described here. The 14 MCATL statements can be grouped into five functions: test division, administration control, scoring, reporting, and customizing.

Figure 1. EBNF Description of MCATL

```
<test>              ::=  "TEST" <name> <term>

                         <statements> "ENDTEST" <term>

<name>              ::=  <1 - 6 letters>

<term>              ::=  <end of line> | "!"

<statements>        ::=  <statement> {<statements>}

<statement>         ::=  [<label> " "]

                         [ <test> |

                           <item statement> |

                           <terminate statement> |

                           <if statement> |

                           <jump statement> |

                           <search statement> |

                           <sequence statement> |

                           <review statement> |
                                <set statement> |
                                <setscore statement> |
                                <keep statement> |
                                <autokeep statement> |
                                <interpret statement> |
                                <custom statement> |
                                <null> ] <term>
<label>                  ::=    "$" <name>
<item statement>         ::=    "#" <item number> [<branch clause>]
                                [<characteristic clause>]
<item number>            ::=    <name> <item specific>
<item specific>          ::=    <1 - 3 digits, not all zero>
<branch clause>          ::=    <correct branch> <incorrect branch>
                                [<error branch>] |
                                <alternative branch> [<error branch> ] |
                                <error branch> |
                                <unconditional branch> [<error branch>]
<correct branch>         ::=    "CO:" <label> |
                                "CORRECT:" <label>
<incorrect branch>       ::=    "IN:" <label> |
                                "INCORRECT:" <label>
<alternative branch>     ::=    ("AL:" | "ALTERNATIVE:") "A=" <label>
                                [ "B=" <label> [ "C=" <label> [ "D=" <label>
                                [ "E=" <label> [ "F=: <label> ] ] ] ] ]
<error branch>           ::=    "ER:" <label> |
                                "ERROR:" <label>
```

```
<unconditional branch> ::= "PROCEED:" <label>|
                           "PR:" <label>
<characteristic clause> ::= <characteristic> [<characteristic cause>]
<characteristic>        ::= "SLIMIT:" <value> |
                            "RLIMIT:" <value> |
                            ("CLEAR" | "NOCLEAR") |
                            "KEY:" <value>
<terminate statement>   ::= "TERMINATE" [<label>] <logical expression>
<logical expression>    ::= ["NOT"] ["("] <relationship> <logical operator>
                            (<relationship> | <logical expression>) [")"]
<relationship>          ::= <arithmetic expression> <relational operator>
                            <arithmetic expression>
<logical operator>      ::= <"AND" | "OR">
<relational operator>   ::= "<" | ">" | "=" | "<=" | ">=" | "<>"
<if statement>          ::= "IF" <logical expression> <term> <statements>
                            {"ELSEIF" [<logical expression>] <term>
                            <statements>} "ENDIF"
<jump statement>        ::= "JUMP <label> [<logical expression>] <term>
<search statement>      ::= "SEARCH" <value> <term>
                            <item statement> {<item statement>}
                            "ENDSEARCH"
<value>                 ::= <var> | <con>
<var>                   ::= ["@"] <letter> <0 - 9 letters or digits>
<con>                   ::= <integer or real number>
<sequence statement>    ::= "SEQUENCE" [branch clause] [ "ONEND:" <label> ]
                            [<characteristic clause>] <term> <item statement>
```

```
                             <term> {<item statement> <term>}
                             "ENDSEQUENCE"
<review statement>       ::= "REVIEW" {<review option>}
<review option>          ::= "ALL" | "SKIPPED" | "SPECIFIC"
<doreview statement>     ::= "DOREVIEW"
<set statement>          ::= "SET" <var> "=" <arithmetic expression>
<expression>             ::= "(" <logical expression> |
                             <arithmetic expression> ")"
<arithmetic expression>  ::= ["("] {"-"} <value> [<arithmetic operator>
                             (<value> | <arithmetic expression>)] [")"]
<arithmetic operator>    ::= "+" | "-" | "*" | "/"
<setscore statement>     ::= "SETSCORE" <scorelist>
<scorelist>              ::= <score function> "<" <value list> ")"
                             {<score function> "(" <value list> ")"}
<score function>         ::= "BANK-IDENTIFIER" |
                             "BAYESIAN" |
                             "CSCORE-1:" | ... | "CSCORE-9" |
                             "KEY" |
                             "LATENCY" |
                             "MAXIMUM-LIKELIHOOD" |
                             "MODAL-BAYESIAN" |
                             "NUMBER-ADMINISTERED" |
                             "NUMBER-CORRECT" |
                             "PROPORTION-CORRECT" |
                             "RESPONSE" |
                             "TIME"
```

```
<value list>            ::=  {<value>}

<keep statement>        ::=  "KEEP" <write list>

<write list>            ::=  (<value> | <string>) {write list}

<string>                ::=  """ <alphabetic character>
                             {<alphabetic character>} """

<autokeep statement>    ::=  "AUTOKEEP" <write list>

<interpret statement>   ::=  "INTERPRET" "(" <interpret clause>
                             {<interpret clause>} ")"

<interpret clause>      ::=  <string> | "^" | <var> | <label>

<custom statement>      ::=  "PROC-1" | ... | "PROC-5"
```

Test Division

The entire MCATL test specification is a test statement. Subtests within the main test are also test statements. A test statement consists of the word TEST, a test name, a group of test specification statements, and the word ENDTEST. All scoring functions are local to the test in which they are listed. Similarly, variables are local to the test unless they are explicitly declared as global. Thus, any number of independent subtests can be included within the main test. Variable names and scoring functions can be reused within each test without interference from other tests.

Subtests can be nested within tests and other subtests to an arbitrary degree. However, this nesting is not hierarchical, as it is in Pascal, because nested subtests are independent of the tests in which they are nested, except for the global variables they contain. Whereas a nested Pascal procedure has access to all variables declared in superordinate procedures, MCATL subtests do not. Nesting in MCATL is available only to allow the administration of independent subtests before the completion of a current test.

Administration Control

Eight statements make up the administration control category. The item statement specifies that an item should be administered or included in a structure. An item statement consists of a # sign, an item number and, optionally, a branch clause and a characteristic clause. The branch clause to the item statement specifies where the item should branch on the specified conditions. The characteristic clause can override certain characteristics of the item, such as whether the screen should be cleared before the item is presented. The branch clause is provided to allow inter-item branching strategies. The characteristic clause override was provided as a convenience, but it has been of little practical use and was therefore not documented in the user's manual. It may be deleted from later versions of the system.

The TERMINATE statement specifies the conditions under which testing should stop. It consists of the word TERMINATE, an optional label to branch to upon termination, and a logical expression that describes the condition under which termination should occur. The TERMINATE expression is evaluated after each item that accepts a response is administered. Typically, only the administration of an item will change the variables contained in a TERMINATE expression. The label was included as an afterthought. Typically, a score is written to the output file after the termination criterion is satisfied. Without the label, the flow of control in the test specification transferred to the ENDTEST. Local variables disappeared, and a score was available to write out only if it was stored in a global variable. The label field allowed for the transfer of control to an output statement, typically near the end of the test.

The IF statement provides a structure that is useful for non-reentrant inter-subtest branching. It begins with the word IF and a logical expression describing the conditions under which the first subtest will be administered. Additional subtests are separated from the first subtest by ELSEIF clauses, which are similar to the original IF clause except that the word ELSEIF is substituted. The IF statement ends with the word ENDIF. Although the initial IF statement requires a logical expression, the logical expression is optional for the ELSEIF clauses. If the logical expression is omitted, the ELSEIF clause will be executed unconditionally. Only the first IF or ELSEIF clause whose logical expression is satisfied will be executed. Thus the IF and all ELSEIFs except the last one should be accompanied by a logical expression. The use of the IF-ELSEIF-ENDIF structure is not limited to the selection of tests, however. It can also be used to separate any collection of statements. However, control cannot be transferred into or out of the IF-ELSEIF-ENDIF structure.

The JUMP statement provides a convenient unstructured alternative to the IF structure for progressing through a test. It is intended for applications for which the structured statements are too cumbersome. When it is used without a logical expression, it provides an unconditional transfer of control. With a logical expression, it provides a conditional transfer of control. It cannot be used to branch outside of the test block in which it appears nor can it be used to branch into the middle of a statement (e.g., an IF statement).

The SEARCH statement initiates a model-based testing strategy. Given a value to search on, it chooses the unadministered item that has the most psychometric information at that value. If the value provided is a score, the search implements an adaptive test. If the value provided is a constant, it administers a conventional test. The searched test will continue until all items have been administered or until the termination criterion specified in the TERMINATE statement has been satisfied.

The SEQUENCE statement is used for reentrant inter-subtest branching. A sequence contains a set of items. Each time the sequence is executed, the next item in the sequence is administered. Branching occurs from the SEQUENCE statement after the item is administered. An ONEND branch is provided for the condition in which the items in the sequence are exhausted. If testing exhausts a sequence that has no ONEND branch, control falls through the ENDSEQUENCE to the next statement in the specification.

The REVIEW statement allows examinees to review items in one of three ways. The examinee can review all of the items, those items he or she skipped, or specific items identified by a sequence number. If the review option is in effect, the review process will automatically be explained at the end of the test and the examinee will be given the opportunity to use it. When a response is changed, all scores are recalculated. If a score is sequence-dependent, the new response will be substituted for the old one. Thus, the score will be the same as if the response given in the review had been given initially.

The final administration control statement is the DOREVIEW statement. The review process is normally initiated when the ENDTEST statement is reached. Often, however, it must be initiated before the ENDTEST (for example, to write post-review scores to the data file). Like the ENDTEST, the DOREVIEW statement initiates the review process, but since the test does not end, all local variables are still available after the review.

Scoring

The SET statement is a simple assignment statement with which the value of an expression can be assigned to a variable. It can be used to initialize score variables and to transform scores.

The SETSCORE statement selects the scoring functions that are to be used in a test or subtest and assigns constants or variable names to the parameters of the scoring functions. A scoring function can be listed more than once in a SETSCORE statement, but variable names cannot. The SETSCORE statement only sets scores up to be calculated. No scores are calculated until the variables assigned as parameters are used.

Reporting

Three reporting statements are provided: the KEEP statement, the AUTOKEEP statement, and the INTERPRET statement. The KEEP statement is a simple file-write statement that causes variables and text to be written to a permanent file, XTESTEEX.KEE. The KEEP statement writes a line to the file every time it is executed in the test specification.

The AUTOKEEP statement is similar to the KEEP statement but it is executed every time an item is administered and a response is accepted. The AUTOKEEP statement is particularly useful when intermediate results, such as item responses, are to be kept in a model-based branching strategy using the SEARCH statement. It also provides a convenient way of keeping response data even if the SEARCH statement is not used.

The INTERPRET statement provides a means of writing a narrative interpretation that can be given to the examinee to explain the test results. It is similar in function to the KEEP statement but provides formatting capabilities for extended text production. It also writes to a separate file, XTESTEEX.INT. In operational use, the KEEP and AUTOKEEP statements are usually used to keep data, and the INTERPRET statement is used to provide a report for the examinee.

Customizing

The five custom statements branch to one of five possible custom processing procedures written by the test developer in a programming language such as Pascal or FORTRAN. When executed, these statements transfer control to the user's procedure, which is responsible for all processing and for the return of control to the testing system when it is done.

Example

Figure 2 shows an example of a test specification. The TEST statement names the test SAMPLE. The SETSCORE statement indicates that two scores are to be computed. The first one, the Bayesian score, uses the variables MEAN and VAR for the posterior means and variances. The prior mean and variance are set at zero and one, respectively. The variable NUMADMIN is assigned to the number-administered scoring function and will contain a count of the number of items that have been administered.

The TERMINATE statement establishes the conditions for terminating the test and identifies the statement to branch to when the conditions are satisfied. This test will terminate when the Bayesian posterior variance becomes less than 0.1 and more than five items have been administered. Upon termination, control will transfer to the statement labelled $TERMOK.

The REVIEW option is activated by the REVIEW statement. In this example, the examinee will be allowed to review items that he or she skipped on the first pass.

A flexilevel testing strategy is implemented in Test SAMPLE. In this example, items are ordered by difficulty. Item 6 is the easiest and Item 15 is the hardest. The strategy is implemented using two SEQUENCE statements labeled SEQ1 and SEQ2. In sequence one, items start at a medium difficulty and become easier further into the sequence. In sequence two, the items again start at a medium difficulty, but become more difficult. Testing begins in sequence one. Following a correct response to sequence one, testing branches to sequence two. After an incorrect response to sequence one, the next item in sequence one is chosen. When sequence one runs out of items, the testing process branches to the label $HITEND. The branching instructions for sequence two are the same as for sequence one. As a result of these branching instructions, examinees who answer a majority of the items correctly will be given more items from sequence two, while examinees who answer a majority incorrectly will be given more items from sequence one.

Figure 2. Test Specification

```
TEST SAMPLE

SETSCORE BAYESIAN(MEAN, VAR, 0, 1),&
NUMBER-ADMINISTERED(NUMADMIN)

TERMINATE $TERMOK ((VAR < 0.1) AND (NUMADMIN > 5))

REVIEW SKIPPED

$SEQ1  SEQUENCE CORRECT: $SEQ2, INCORRECT: $SEQ1,&
ONEND: $HITEND

#ITEM10
#ITEM9
#ITEM8
#ITEM7
#ITEM6
ENDSEQUENCE

$SEQ2  SEQUENCE CORRECT: $SEQ2, INCORRECT: $SEQ1,&
ONEND: $HITEND

#ITEM11
#ITEM12
#ITEM13
#ITEM14
#ITEM15
ENDSEQUENCE

$HITEND KEEP ("TEST TERMINATED BECAUSE IT RAN OUT OF
ITEMS AFTER ", NUMADMIN, " HAD BEEN GIVEN.")

$TERMOK KEEP ("THE SCORE BEFORE REVIEW WAS ", MEAN)

DOREVIEW

KEEP ("THE SCORE AFTER REVIEW WAS ", MEAN)

ENDTEST
```

The KEEP statement labeled $HITEND inserts a message into the permanent data file indicating that the test terminated because it ran out of items. The KEEP statement labeled $TERMOK inserts a line in the permanent data file indicating the Bayesian posterior mean before any items are reviewed.

The DOREVIEW statement initiates the review process. When this statement is executed, the examinee is given the option of reviewing the items that he or she skipped during the first pass.

The KEEP statement following the DOREVIEW statement writes the final score to the permanent file. It says, "The score after review was", and prints the value of the Bayesian posterior mean. This score is calculated using the new responses scored in the order of the original item presentation.

The ENDTEST ends the sample test. If the DOREVIEW statement had not been included within the test, the ENDTEST statement would execute the review.

LANGUAGE IMPLEMENTATION AND EVALUATION

A compiler for MCATL was implemented on an IBM PC as part of the MicroCAT Testing System. It compiles the MCATL specification to an intermediate code from which the test can be administered, and at the same time performs as many of the computations as possible that will be needed for administration. Test items are randomly accessible in the test file. In addition, graphic test items, which are stored in the item banks as strings of graphic commands, are translated into pixel representation and compressed for rapid presentation during testing. Furthermore, search tables are set up so that information-based item selection can proceed rapidly with only a table look-up and no computations during the testing session.

In the MicroCAT implementation, an MCATL pre-processor was developed to provide a simpler, more user-friendly interface for the test developer who is relatively unfamiliar with programming languages. This pre-processor creates MCATL test specifications from a test template and user-supplied information. An MCATL template typically consists of an MCATL test specification with several blanks for the user to fill in. These blanks may be filled with item reference numbers, termination criteria, etc. In addition to the MCATL statements and the blanks, INSTRUCT statements are usually included in the template. The INSTRUCT statements are displayed by the pre-processor as they are encountered and are used to inform the user what information the pre-processor is requesting. The output of the MCATL pre-processor is an MCATL test specification. Like any MCATL test specification, this has to be compiled before the test can be administered.

The MCATL compiler of the MicroCAT system has been tested internally for nearly two years, and it has been field tested for more than one year. Although the language and the compiler have not been formally evaluated, first users have indicated that they are pleased with its power and simplicity in setting up adaptive tests. Three levels of involvement were provided with the system: the template level, the MCATL level, and the Pascal level. All three levels are currently being used. The template level provides very fast development of tests using standard strategies. The MCATL level allows specification of the more common adaptive testing strategies. The Pascal level has provided a necessary extension for some new adaptive strategies that were not envisioned at the time the language was designed and are not yet commonplace enough to warrant their inclusion in the language.

Perhaps the most noticeable shortcoming of MCATL is its occasional tendency to provide features that are not useful. One example of this may be the test nesting capability. MCATL was designed to allow several levels of nesting of subtests within tests. However, this is rarely done; in fact, tests are usually not nested at all. The first test may consist of instructional items, the second test may consist of measurement items, and a third test may consist of feedback items. These tests are independent, however, and do not need to be nested. Similarly, the capability to override item characteristics in the item statement was included in the original specification. In fact, this will probably never be done. This feature was not documented in the MicroCAT manual and may be removed from future versions of the language. The advantage of eliminating these extraneous capabilities will be a reduction in size and an enhancement in speed of the testing system.

In general, MCATL has functioned well. The separate compilation phase has proved very advantageous in terms of execution speed. Whereas it may take as much as one minute per item to compile items with very complex graphics, the time required to display the item during execution is uniformly about half a second. Similarly, although it may take several minutes to construct a search table for a mathematically branched testing strategy, the time it takes to retrieve an item during testing is uniformly less than two seconds, regardless of the testing strategy used.

REFERENCES

Assessment Systems Corporation. (1984). *User's manual for the MicroCAT Testing System* (ONR-85-1). St. Paul, MN: Author.

Jensen, K., & Wirth, N. (1974). *Pascal user manual and report.* New York: Springer-Verlag.

Vale, C. D. (1981). Design and implementation of a microcomputer-based adaptive testing system. *Behavior Research Methods and Instrumentation, 13*(4), 399-406.

Weiss, D. J. (Ed.) (1978). *Proceedings of the 1977 Computerized Adaptive Testing Conference.* Minneapolis: University of Minnesota, Department of Psychology, Psychometric Methods Program.

Weiss, D. J. (Ed.) (1980). *Proceedings of the 1979 Computerized Adaptive Testing Conference.* Minneapolis: University of Minnesota, Department of Psychology, Psychometric Methods Program.

Weiss, D. J. (Ed.) (1985). *Proceedings of the 1982 Item Response Theory and Computerized Adaptive Testing Conference.* Minneapolis: University of Minnesota, Department of Psychology, Psychometric Methods Program.

Distribution List

Personnel Analysis Division,
AF/MPXA
5C360, The Pentagon
Washington, DC 20330

Air Force Human Resources Lab
AFHRL/MPD
Brooks AFB, TX 78235

Dr. Earl A. Alluisi
HQ, AFHRL (AFSC)
Brooks AFB, TX 78235

Dr. Erling B. Andersen
Department of Statistics
Studiestraede 6
1455 Copenhagen
DENMARK

Dr. Phipps Arabie
University of Illinois
Department of Psychology
603 E. Daniel St.
Champaign, IL 61820

Technical Director, ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Eva L. Baker
UCLA Center for the Study
of Evaluation
145 Moore Hall
University of California
Los Angeles, CA 90024

Dr. Isaac Bejar
Educational Testing Service
Princeton, NJ 08450

Dr. Menucha Birenbaum
School of Education
Tel Aviv University
Tel Aviv, Ramat Aviv 69978
ISRAEL

Dr. Arthur S. Blaiwes
Code N711
Naval Training Equipment Center
Orlando, FL 32813

Dr. R. Darrell Bock
University of Chicago
Department of Education
Chicago, IL 60637

Cdt. Arnold Bohrer
Sectie Psychologisch Onderzoek
Rekruterings-En Selectiecentrum
Kwartier Koningen Astrid
Bruijnstraat
1120 Brussels, BELGIUM

Dr. Robert Breaux
Code N-095R
NAVTRAEQUIPCEN
Orlando, FL 32813

Dr. Robert Brennan
American College Testing
Programs
P. O. Box 168
Iowa City, IA 52243

Dr. Patricia A. Butler
NIE Mail Stop 1806
1200 19th St., NW
Washington, DC 20208

Mr. James W. Carey
Commandant (G-PTE)
U.S. Coast Guard
2100 Second Street, S.W.
Washington, DC 20593

Dr. James Carlson
American College Testing
Program
P.O. Box 168
Iowa City, IA 52243

Dr. John B. Carroll
409 Elliott Rd.
Chapel Hill, NC 27514

Dr. Robert Carroll
NAVOP 01B7
Washington, DC 20370

Dr. Norman Cliff
Department of Psychology
Univ. of So. California
University Park
Los Angeles, CA 90007

Distribution List (Continued)

Director,
Manpower Support and
Readiness Program
Center for Naval Analysis
2000 North Beauregard Street
Alexandria, VA 22311

Dr. Stanley Collyer
Office of Naval Technology
Code 222
800 N. Quincy Street
Arlington, VA 22217-5000

Dr. Hans Crombag
University of Leyden
Education Research Center
Boerhaavelaan 2
2334 EN Leyden
The NETHERLANDS

CTB/McGraw-Hill Library
2500 Garden Road
Monterey, CA 93940

Dr. Dattprasad Divgi
Center for Naval Analysis
4401 Ford Avenue
P.O. Box 16268
Alexandria, VA 22302-0268

Dr. Hei-Ki Dong
Ball Foundation
800 Roosevelt Road
Building C, Suite 206
Glen Ellyn, IL 60137

Defense Technical
Information Center
Cameron Station, Bldg 5
Alexandria, VA 22314
Attn: TC
(12 Copies)

Dr. Stephen Dunbar
Lindquist Center
for Measurement
University of Iowa
Iowa City, IA 52242

Dr. James A. Earles
Air Force Human Resources Lab
Brooks AFB, TX 78235

Dr. Kent Eaton
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. John M. Eddins
University of Illinois
252 Engineering Research
Laboratory
103 South Mathews Street
Urbana, IL 61801

Dr. Susan Embretson
University of Kansas
Psychology Department
Lawrence, KS 66045

ERIC Facility-Acquisitions
4833 Rugby Avenue
Bethesda, MD 20014

Dr. Benjamin A. Fairbank
Performance Metrics, Inc.
5825 Callaghan
Suite 225
San Antonio, TX 78228

Dr. Leonard Feldt
Lindquist Center
for Measurement
University of Iowa
Iowa City, IA 52242

Dr. Richard L. Ferguson
American College Testing
Program
P.O. Box 168
Iowa City, IA 52240

Dr. Gerhard Fischer
Liebiggasse 5/3
A 1010 Vienna
AUSTRIA

Prof. Donald Fitzgerald
University of New England
Department of Psychology
Armidale, New South Wales 2351
AUSTRALIA

Mr. Paul Foley
Navy Personnel R&D Center
San Diego, CA 92152

Distribution List (Continued)

Dr. Carl H. Frederiksen
McGill University
3700 McTavish Street
Montreal, Quebec H3A 1Y2
CANADA

Dr. Robert D. Gibbons
University of Illinois-Chicago
P.O. Box 6998
Chicago, IL 69680

Dr. Janice Gifford
University of Massachusetts
School of Education
Amherst, MA 01003

Dr. Robert Glaser
Learning Research
& Development Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Bert Green
Johns Hopkins University
Department of Psychology
Charles & 34th Street
Baltimore, MD 21218

Dr. Ronald K. Hambleton
Prof. of Education & Psychology
University of Massachusetts
at Amherst
Hills House
Amherst, MA 01003

Ms. Rebecca Hetter
Navy Personnel R&D Center
Code 62
San Diego, CA 92152

Dr. Paul W. Holland
Educational Testing Service
Rosedale Road
Princeton, NJ 08541

Prof. Lutz F. Hornke
Universitat Dusseldorf
Erziehungswissenschaftliches
Universitatsstr. 1
Dusseldorf 1
WEST GERMANY

Dr. Paul Horst
677 G Street, #184
Chula Vista, CA 90010

Mr. Dick Hoshaw
NAVOP-135
Arlington Annex
Room 2834
Washington, DC 20350

Dr. Lloyd Humphreys
University of Illinois
Department of Psychology
603 East Daniel Street
Champaign, IL 61820

Dr. Steven Hunka
Department of Education
University of Alberta
Edmonton, Alberta
CANADA

Dr. Huynh Huynh
College of Education
Univ. of South Carolina
Columbia, SC 29208

Dr. Robert Jannarone
Department of Psychology
University of South Carolina
Columbia, SC 29208

Dr. Douglas H. Jones
Advanced Statistical
Technologies Corporation
10 Trafalgar Court
Lawrenceville, NJ 08148

Dr. G. Gage Kingsbury
Portland Public Schools
Research and Evaluation Department
501 North Dixon Street
P. O. Box 3107
Portland, OR 97209-3107

Dr. William Koch
University of Texas-Austin
Measurement and Evaluation
Center
Austin, TX 78703

Distribution List (Continued)

Dr. Leonard Kroeker
Navy Personnel R&D Center
San Diego, CA 92152

Dr. Michael Levine
Educational Psychology
210 Education Bldg.
University of Illinois
Champaign, IL 61801

Dr. Charles Lewis
Faculteit Sociale Wetenschappen
Rijksuniversiteit Groningen
Oude Boteringestraat 23
9712GC Groningen
The NETHERLANDS

Dr. Robert Linn
College of Education
University of Illinois
Urbana, IL 61801

Dr. Robert Lockman
Center for Naval Analysis
4401 Ford Avenue
P.O. Box 16268
Alexandria, VA 22302-0268

Dr. Frederic M. Lord
Educational Testing Service
Princeton, NJ 08541

Dr. James Lumsden
Department of Psychology
University of Western Australia
Nedlands W.A. 6009
AUSTRALIA

Dr. William L. Maloy
Chief of Naval Education
and Training
Naval Air Station
Pensacola, FL 32508

Dr. Gary Marco
Stop 31-E
Educational Testing Service
Princeton, NJ 08451

Dr. Clessen Martin
Army Research Institute
5001 Eisenhower Blvd.
Alexandria, VA 22333

Dr. James McBride
Psychological Corporation
c/o Harcourt, Brace,
Javanovich Inc.
1250 West 6th Street
San Diego, CA 92101

Dr. Clarence McCormick
HQ, MEPCOM
MEPCT-P
2500 Green Bay Road
North Chicago, IL 60064

Mr. Robert McKinley
University of Toledo
Department of Educational Psychology
Toledo, OH 43606

Dr. Barbara Means
Human Resources
Research Organization
1100 South Washington
Alexandria, VA 22314

Dr. Robert Mislevy
Educational Testing Service
Princeton, NJ 08541

Headquarters, Marine Corps
Code MPI-20
Washington, DC 20380

Dr. W. Alan Nicewander
University of Oklahoma
Department of Psychology
Oklahoma City, OK 73069

Dr. William E. Nordbrock
FMC-ADCO Box 25
APO, NY 09710

Dr. Melvin R. Novick
356 Lindquist Center
for Measurement
University of Iowa
Iowa City, IA 52242

Director, Manpower and Personnel
Laboratory,
NPRDC (Code 06)
San Diego, CA 92152

Distribution List (Continued)

Library, NPRDC
Code P201L
San Diego, CA 92152

Commanding Officer,
Naval Research Laboratory
Code 2627
Washington, DC 20390

Dr. James Olson
WICAT, Inc.
1875 South State Street
Orem, UT 84057

Office of Naval Research,
Code 1142PT
800 N. Quincy Street
Arlington, VA 22217-5000
(6 Copies)

Special Assistant for Marine
Corps Matters,
ONR Code 00MC
800 N. Quincy St.
Arlington, VA 22217-5000

Dr. Judith Orasanu
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Wayne M. Patience
American Council on Education
GED Testing Service, Suite 20
One Dupont Circle, NW
Washington, DC 20036

Dr. James Paulson
Department of Psychology
Portland State University
P.O. Box 751
Portland, OR 97207

Dr. Roger Pennell
Air Force Human Resources
Laboratory
Lowry AFB, CO 80230

Dr. Mark D. Reckase
ACT
P. O. Box 168
Iowa City, IA 52243

Dr. Malcolm Ree
AFHRL/MP
Brooks AFB, TX 78235

Dr. Carl Ross
CNET-PDCD
Building 90
Great Lakes NTC, IL 60088

Dr. J. Ryan
Department of Education
University of South Carolina
Columbia, SC 29208

Dr. Fumiko Samejima
Department of Psychology
University of Tennessee
Knoxville, TN 37916

Mr. Drew Sands
NPRDC Code 62
San Diego, CA 92152

Dr. Robert Sasmor
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Mary Schratz
Navy Personnel R&D Center
San Diego, CA 92152

Dr. W. Steve Sellman
OASD(MRA&L)
2B269 The Pentagon
Washington, DC 20301

Dr. Kazuo Shigemasu
7-9-24 Kugenuma-Kaigan
Fujusawa 251
JAPAN

Dr. William Sims
Center for Naval Analysis
4401 Ford Avenue
P.O. Box 16268
Alexandria, VA 22302-0268

Distribution List (Continued)

Dr. H. Wallace Sinaiko
Manpower Research
and Advisory Services
Smithsonian Institution
801 North Pitt Street
Alexandria, VA 22314

Dr. Richard Sorensen
Navy Personnel R&D Center
San Diego, CA 92152

Dr. Paul Speckman
University of Missouri
Department of Statistics
Columbia, MO 65201

Dr. Martha Stocking
Educational Testing Service
Princeton, NJ 08541

Dr. Peter Stoloff
Center for Naval Analysis
200 North Beauregard Street
Alexandria, VA 22311

Dr. William Stout
University of Illinois
Department of Mathematics
Urbana, IL 61801

Maj. Bill Strickland
AF/MPXOA
4E168 Pentagon
Washington, DC 20330

Dr. Hariharan Swaminathan
Laboratory of Psychometric and
Evaluation Research
School of Education
University of Massachusetts
Amherst, MA 01003

Mr. Brad Sympson
Navy Personnel R&D Center
San Diego, CA 92152

Dr. Kikumi Tatsuoka
CERL
252 Engineering Research
Laboratory
Urbana, IL 61801

Dr. Maurice Tatsuoka
220 Education Bldg
1310 S. Sixth St.
Champaign, IL 61820

Dr. David Thissen
Department of Psychology
University of Kansas
Lawrence, KS 66044

Mr. Gary Thomasson
University of Illinois
Educational Psychology
Champaign, IL 61820

Dr. Robert Tsutakawa
The Fred Hutchinson
Cancer Research Center
Division of Public Health Sci.
1124 Columbia Street
Seattle, WA 98104

Dr. Ledyard Tucker
University of Illinois
Department of Psychology
603 E. Daniel Street
Champaign, IL 61820

Dr. Vern W. Urry
Personnel R&D Center
Office of Personnel Management
1900 E. Street, NW
Washington, DC 20415

Dr. David Vale
Assessment Systems Corp.
2233 University Avenue
Suite 310
St. Paul, MN 55114

Dr. Frank Vicino
Navy Personnel R&D Center
San Diego, CA 92152

Dr. Howard Wainer
Division of Psychological Studies
Educational Testing Service
Princeton, NJ 08541

Distribution List (Continued)

Dr. Ming-Mei Wang
Lindquist Center
for Measurement
University of Iowa
Iowa City, IA 52242

Mr. Thomas A. Warm
Coast Guard Institute
P. O. Substation 18
Oklahoma City, OK 73169

Dr. Brian Waters
Program Manager
Manpower Analysis Program
HumRRO
1100 S. Washington St.
Alexandria, VA 22314

Dr. David J. Weiss
N660 Elliott Hall
University of Minnesota
75 E. River Road
Minneapolis, MN 55455

Dr. Ronald A. Weitzman
NPS, Code 54Wz
Monterey, CA 92152

Major John Welsh
AFHRL/MOAN
Brooks AFB, TX 78223

Dr. Rand R. Wilcox
University of Southern
California
Department of Psychology
Los Angeles, CA 90007

German Military Representative
ATTN: Wolfgang Wildegrube
Streitkraefteamt
D-5300 Bonn 2
4000 Brandywine Street, NW
Washington, DC 20016

Dr. Bruce Williams
Department of Educational
Psychology
University of Illinois
Urbana, IL 61801

Dr. Hilda Wing
Army Research Institute
5001 Eisenhower Ave.
Alexandria, VA 22333

Dr. Martin F. Wiskoff
Navy Personnel R & D Center
San Diego, CA 92152

Mr. John H. Wolfe
Navy Personnel R&D Center
San Diego, CA 92152

Dr. George Wong
Biostatistics Laboratory
Memorial Sloan-Kettering
Cancer Center
1275 York Avenue
New York, NY 10021

Dr. Wendy Yen
CTB/McGraw Hill
Del Monte Research Park
Monterey, CA 93940