DOCUMENT RESUME

ED 190 080 IR 008 561

TITLE	Individualized Instruction for Data Access (IIDA). Quarterly Reports No. 2-3.
INSTITUTION	Drexel Univ., Philadelphia, Pa. Graduate School of Library Science.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	78
GRANT	DSI-77-26524
NOTE	69p.: For related documents, see ED 145 826, ED 168 462, and ED 179 195, and also IR 008 562.
EDRS PRICE	MF01/PC03 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; Computer Programs; Data Bases; *Formative Evaluation; *Individualized Instruction; *Information Retrieval; Relevance (Information Retrieval); *Search Strategies; Summative Evaluation
IDENTIFIERS	*Individualized Instruction for Data Access

ABSTRACT

These quarterly reports for September and December 1978 are two in a series providing a formative evaluation of the Individualized Instruction for Data Access (IIDA) project, an investigation of searching behavior patterns conducted to develop (1) a model of good searching procedure which could be used as the basis for teaching new users of IIDA how to search, (2) specific indicators which the IIDA program could use to analyze a search in progress and determine trends of searching behavior, and (3) an analysis of commonly made errors and the means by which they can be detected and corrected by the IIDA program. Report No. 2 contains a search process analysis (including a searching behavior study, diagnostic procedures, and a search exercise), and a search process assessment (including an error analysis and an identification of measures which discriminate between users). Report No. 3 examines three issues: aspects of formative evaluation planning which overlap with summative evaluation, plans for evaluation of the impact of the system on users with IIDA as either assistant or instructor, and kinds of measures which can and should be used in assessing the impact of IIDA. (FM)

INDIVIDUALIZED INSTRUCTION FOR DATA ACCESS (IIDA)

Quarterly Report No. 2
September, 1978

Drexel University, School of Library
and Information Science
Franklin Institute Research Laboratories

NSF Grant No. DSI 77-26524

TABLE OF CONTENTS

I. OVERVIEW

II. SEARCH PROCESS ANALYSIS

1. Searching Behavior Study

1.1 Procedure
1.2 The Search Requests
1.3 Search Study Results
1.4 Pre-Search Problem Solving
1.5 Use of the Model

2. Diagnostic Procedures and a Search Exercise

2.1 Introduction
2.2 Diagnostic Procedures
2.2.1 Search Structure
2.2.2 Syntactic Analysis
2.2.3 Procedural Analysis
2.2.4 Conversations with the User
2.3 Exercise 2
2.3.1 Program Description
2.3.2 Threshold Analyzer
2.3.3 Warning Control Program
2.3.4 Expand Look-up Subroutine

III. SEARCH PROCESS ASSESSMENT

1. Error Analysis

2. Identification of Measures which Discriminate Between Users

2.1 Introduction
2.2 Objectives and Rationale
2.3 Methodology
2.3.1 Subjects
2.3.2 Variables
2.3.3 Search Problems
2.3.4 Procedure
2.4 Results
2.5 Discussion

IV. REFERENCES

Appendix A: On-Line Searching Project Instructions

Appendix B: Search Requests

Appendix C: On-Line Searching Error Classification Definitions

I. OVERVIEW

This project represents a renewal of earlier work on Individualized Instruction for Data Access (IIDA). Begun in July 1976, with initial funding for one year, the project was resumed in April 1978 and is to be completed in two years. This series of quarterly progress reports is planned to report in depth on selected aspects of the project and to contain a brief, overall progress statement in each report.

The project staff are divided into two groups. The computer group is concerned with the design, implementation and testing of the requisite computer programs. The behavioral group is concerned with formative and summative evaluation of IIDA. In formative evaluation of IIDA our concern is with monitoring system development and with providing feedback and information for refinement and further development of the system. In summative evaluation of IIDA our concern is with an assessment of the impact and effectiveness of the system and the extent to which the objectives of the project are met.

Given that the system can not be subjected to summative evaluation until certain basic programming chores have been completed most of our activities have centered around refinement of project design. Consequently the bulk of this report will be focused on formative evaluation with plans for summative evaluation being a major topic in the next report.

One major activity of the behavioral group has been the investigation of searching behavior patterns in order to develop:

1) a model of good searching procedure which could be used as the basis for teaching new users of IIDA how to search (in the exercise mode), and as a means of determining the searching behaviors which should lead to either successful or unsatisfactory search results;

2) specific indicators which the IIDA program could use to analyze a search in progress (in the assistance mode) and determine trends of searching behavior which are likely to produce less than satisfactory results or which are simply non-productive or inefficient; and

3) an analysis of commonly made errors and means by which they can be detected and corrected by the IIDA program.

In order to accomplish these objectives we have been following two lines of investigation. The first of these is an in-depth analysis of the searching behavior of a few professional searchers. The second is an attempt to identify and develop widely applicable measures of searching performance which discriminate among searchers with varying degrees of experience.

The second section of this report will deal with the first two objectives mentioned above and will discuss the conduct and applications of our first line of investigation. The third section of this report will discuss the second line of investigation and deals mainly with the second and third of the two objectives mentioned above. (This latter portion of the report presents the substance of a proposed research project currently being conducted as part of a Ph. D. dissertation by one of the project members.)

II. SEARCH PROCESS ANALYSIS

1. Searching Behavior Study

The assumption which underlies this line of investigation is that a clearer understanding of what a good searcher does when doing a good job of searching should be beneficial in refining IIDA.

1.1 Procedure

In our first attempt at analyzing searching behavior, we utilized a large number of transcripts of searches obtained from local cooperating libraries. These searches were the results of actual user requests and varied in many respects, such as the search systems and data bases used, the search typics, and the scope of the search. Finding an underlying model for searching using these transcripts proved to be very difficult, both because of the great diversity among them, and because the transcripts alone did not always give sufficient information about the problem solving process that the searcher was engaged in during searching. Quite often the reason for inputting a certain command or series of commands was not at all obvious and the search could not be meaningfully evaluated.

To overcome these problems we devised a more controlled study using one search system (Lockheed), one data base (ChemCon '72 - '76), and three search requests which we selected and pilot tested ourselves. We asked nine different searchers, all of whom performed on-line searching as part of their jobs, to do one search each so that each of the search requests was searched by three different searchers. In order to provide more insight into the procedures used to solve search problems, we asked the searchers not only to conduct the search on-line, but also to "think out loud" while formulating their strategy for the search.

This procedure is an adaptation of the protocol method used by Newell and Simon (1) to study general problem-solving behaviors. Although Newell and Simon asked their subjects to "think out loud" during the entire problem solving process, we did not require searchers to talk about their procedure while they were actually on-line. In pilot testing we found that thinking out loud while doing on-line searching was very disruptive for some searchers and could lead to a less efficient search than they normally performed. We found, however, that the transcript itself could substitute very nicely for thinking out loud when it was supplimented with the searcher's thinking out loud before going on-line (during the strategy formulation phase). Furthermore, the transcripts became even more informative when supplimented by the searcher's comments in an interview conducted after the search was complete.

In summary, the procedure we used was to give the searchers a search request and ask them to think out loud (into a tape recorder) while they looked over the request and began to formulate a strategy for solving it. Then the searchers were asked to actually conduct the search on-line. Finally, we asked the searchers to go over the transcript when the search was complete and to verbally describe the process. In particular they were asked to explain their reasons for any changes they had made in their initial strategy as a result of the way the search progressed. The instructions appear in Appendix A.

1.2 The Search Requests

Three different search requests were used for this study in order to determine whether the intended or expected scope of the retrieval set would evoke any major differences in searching behavior. The search requests appear in Appendix B. The three search requests were each structured differently to try to stimulate maximum amounts of particular kinds of searching behaviors. Each was intended to represent one of the types of searches described by Markey and Atherton (2).

The first request was designed to represent a fairly standard search with a moderate sized retrieval set (our pilot test retrieved 39 hits). The request involved the logical conjunction (AND) of two sets created by combining related terms (OR) and thus represented a "building block" type of search. More advanced features peculiar to the ChemCon data base, such as utilizing chemical registry numbers or Chem Abstracts section codes, as searching aids could be used but were not essential for a successful search.

The second request referenced a particular article and asked for more articles on the same subject. This type of request should elicit a "pearl-growing" type of search in which the searcher begins with one hit and progressively expands the retrieval set.

The third request tried to elicit the opposite response--a "successive fractions" type of search. The request specified a fairly large and general subject area (one with over 500 hits) and asked for a highly limited retrieval set ("a few major references"). The appropriate search behavior would be to progressively reduce the number of hits retrieved. (The members of the project have begun to refer to this type of search as "onion peeling.")

1.3 Search Study Results

The search flow in each of the nine searches was analyzed. The searches were compared with each other to look for underlying similarities among the search methods used. Then the three searches for each request were compared in more detail to look for procedural differences attributable to the type of search requested.

The overall comparison of all nine searches revealed some striking similarities in the basic search flow (See Figure 1). With relatively minor variations each search followed a similar four-step recursive pattern: a) strategy formulation; b) selection and combination of terms to form an initial set; c) a decision as to the adequacy of that set (usually made on the basis of viewing some of the records retrieved by the first pass); and d) either recycling through steps a through c or printing out the results of the search. At this level of analysis the basic search flow would appear to be general

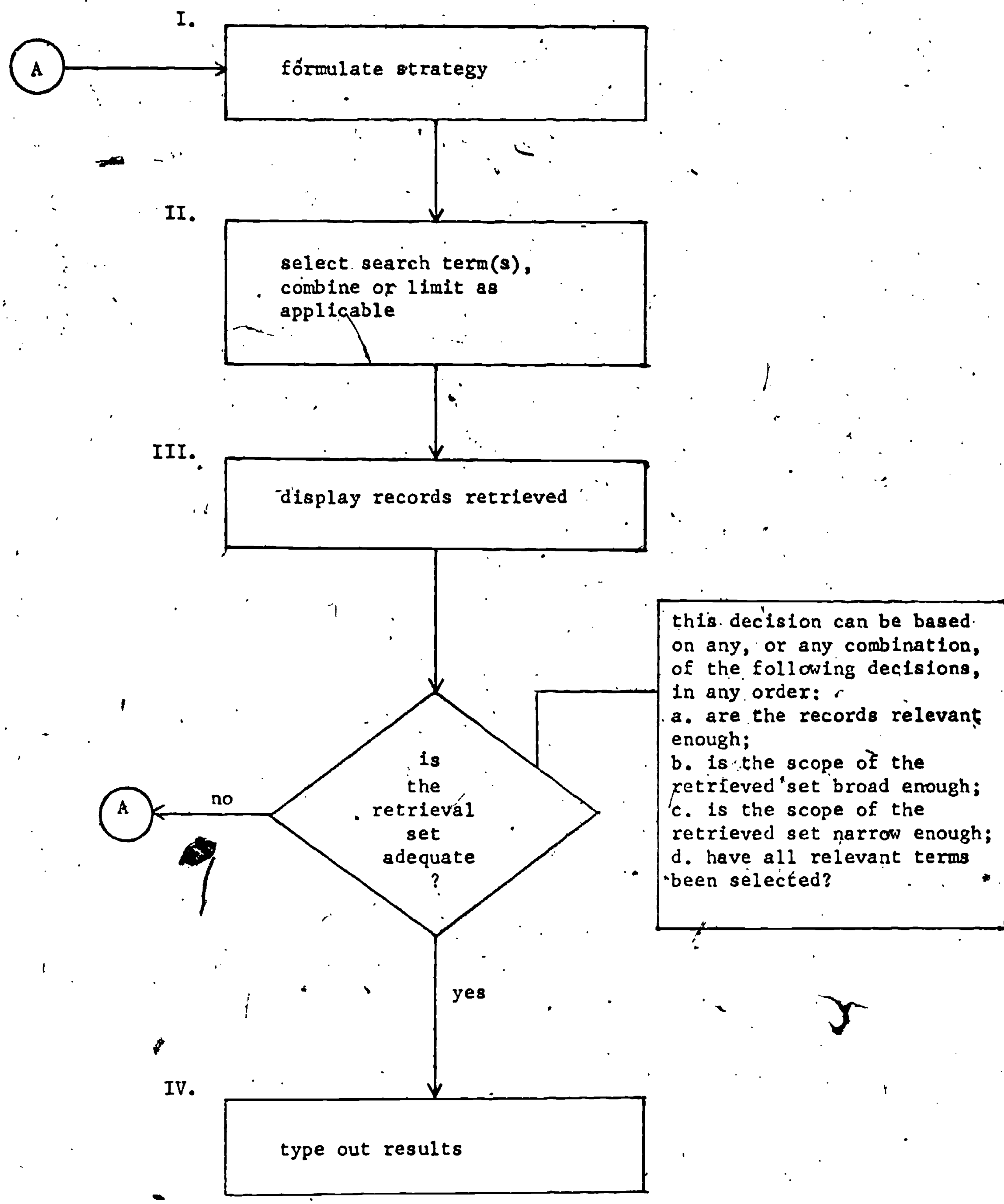

Figure 1. Generalized search flowchart.

not only across searchers and search types but also across search systems. For example, in an article on the use of the SDC system, Morrow (3) presents an essentially identical flowchart.

Analysis of the differences in search behavior as a function of search request types has so far revealed few differences beyond those expected. The first search request, requiring a "building block" approach, successfully elicited this type of behavior. It was also, apparently, the most complex type of the three. Furthermore it prompted the most rigorous pre-search behavior. All pertinent searching aids offered by Chem Abstracts on-line and off-line were utilized by at least one or more of the searchers to determine appropriate search terms. These included synonyms and registry numbers for the compounds named, and section codes and standard terms or abbreviations used for the analytical methods requested. Once the pre-search sources had been exhausted and a satisfactory list of terms compiled the actual search was very straightforward. Terms were selected and related terms were OR'ed together to form two sets. Next these two sets, representing the two major concepts of the search, were AND'ed together to form the final set. Two of the three searchers repeated the process (See Figure 2). In one case the iteration was triggered by an unexpectedly low number of hits for a term. Upon displaying a few of the records from the initial set (step c), the searcher discovered a preferred term which he then selected and combined with the other terms to achieve a final set that seemed satisfactory. In the second case the searcher began with only the most basic terms (a section code number for analytical methods, and registry numbers for the two major compounds named) to see if this simple strategy would suffice before trying a more elaborate one. On receiving a very small set he displayed a few records and noticed that the section code was retreiving some records on methods other than the ones requested. He then reiterated by expanding the compound set to include synonyms and by restricting the methods set by entering keywords for the methods actually requested.

The second search request was not as successful as the first in eliciting the type of search ("pearl growing") it was designed to demonstrate. One of the three searches was sufficiently poor (owing to the searcher's lack of familiarity with the topic) that it has resisted analysis. The other two searches were almost identical, using keywords from the title of the "seed" article to produce a set of other relevant documents (See Figure 3). One of these two used a more restrictive combination of keywords and, upon retrieving a very small set, sought to enlarge the set by recombining them in a less restrictive way. The second search was judged by the searcher to be adequate after the first pass and no iterations were made.

The third search request, like the first, elicited the expected searching behavior. All three of these searchers began by selecting the general term specified in the search topic and then proceeded to limit this set in various ways until, in the searcher's estimation, the set size was small enough to fit the requirements of the request (See Figure 4).

1.4 Pre-Search Problem Solving

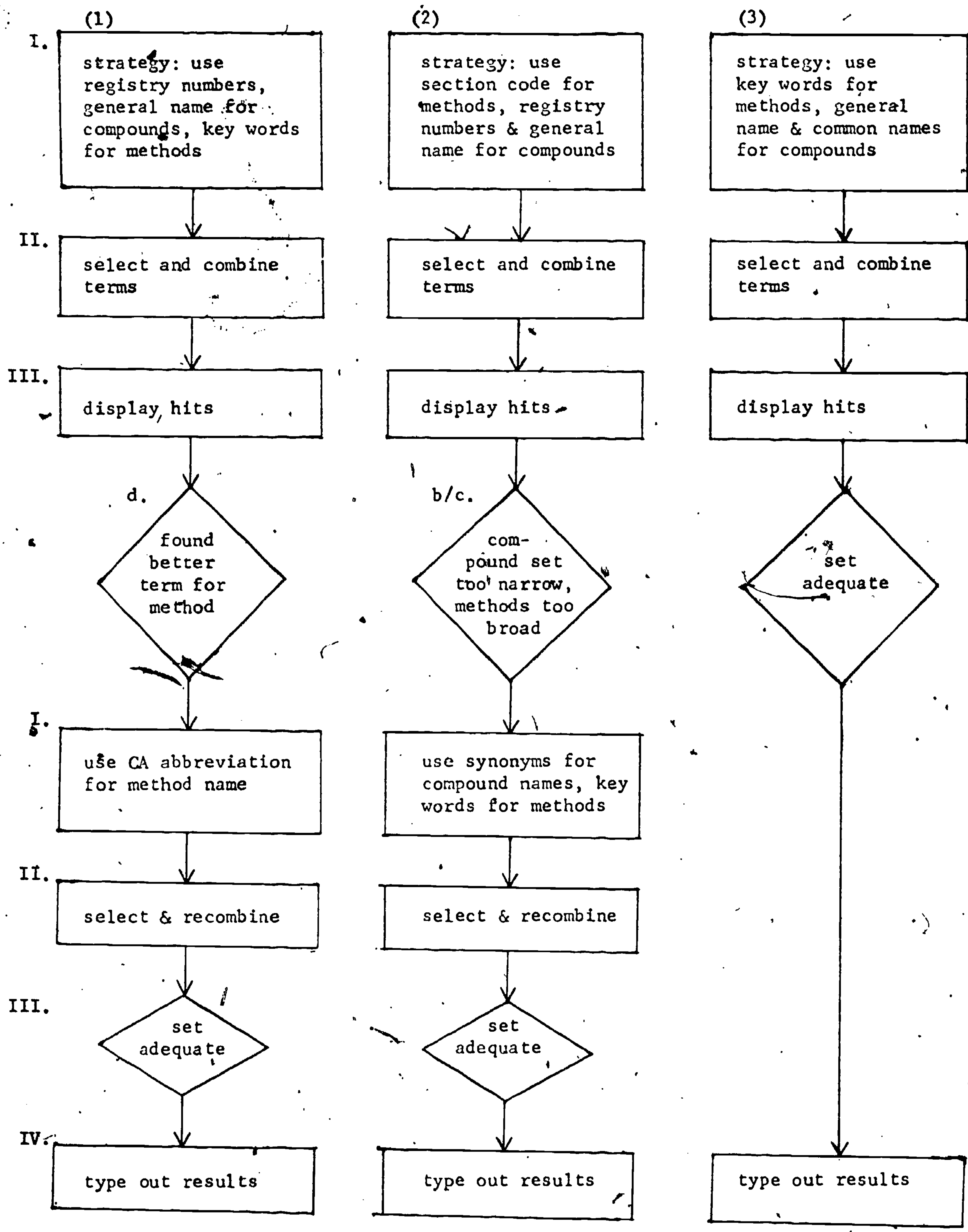

Figure 2. Flow charts for search request 1.

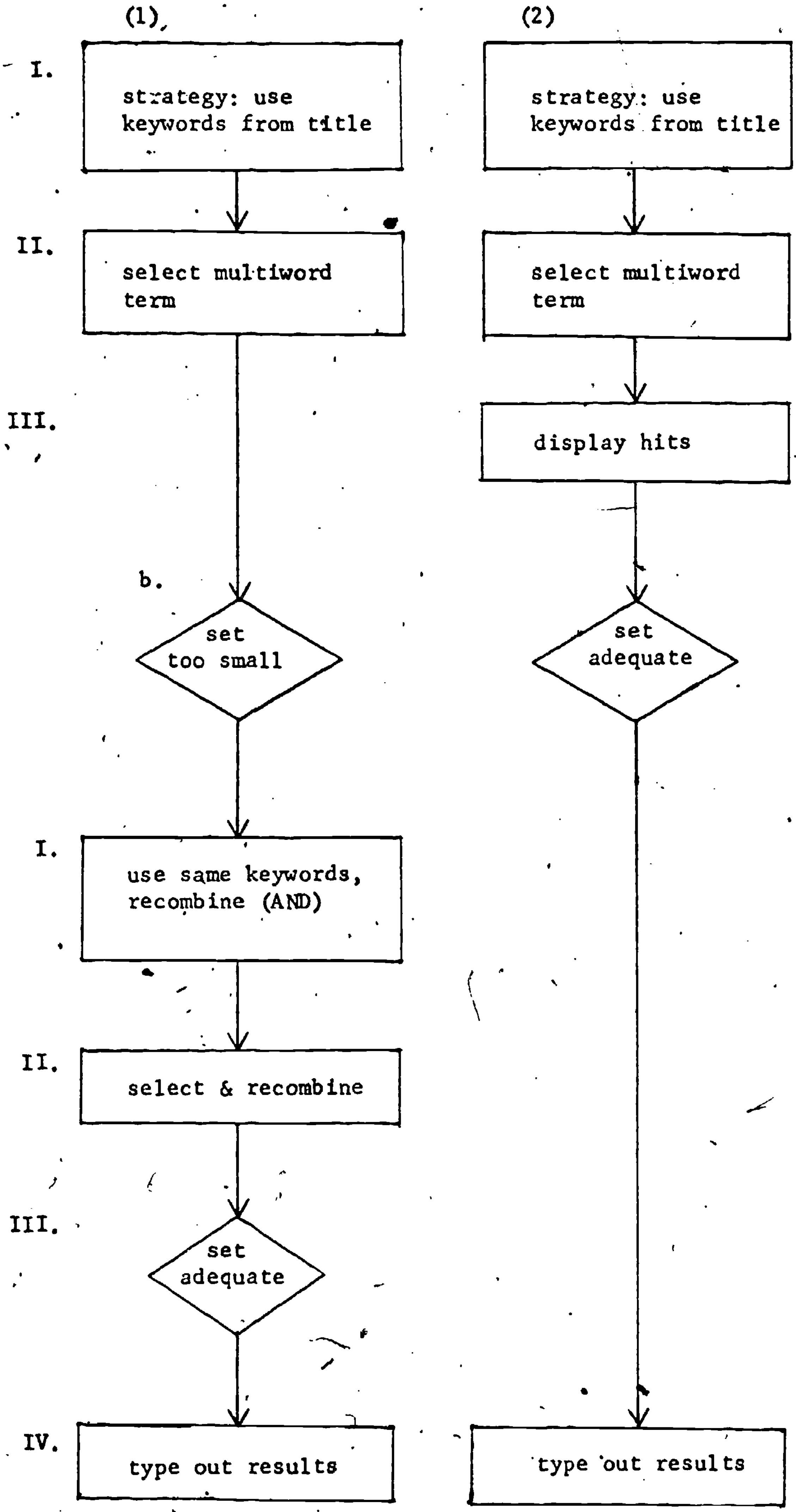

Figure 3. Flow charts for search request 2.

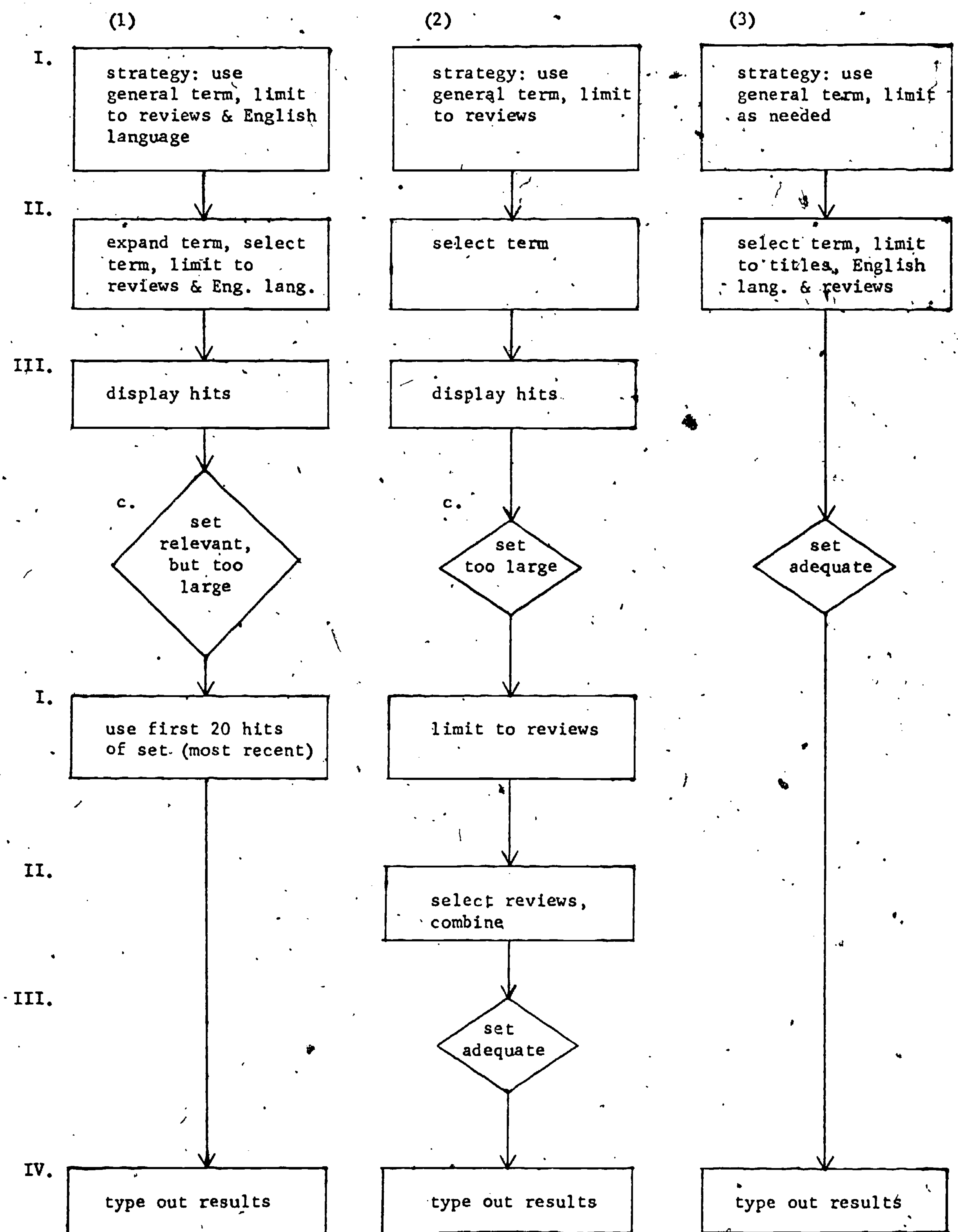

Figure 4. Flow charts for search request 3.

Another area of search behavior analysis we are attempting with the experimental searches is an attempt to construct problem-solving graphs for the searchers' pre-search strategy formulation phase similar to those developed by Newell and Simon (1) for some of the problems which they studied. This procedure involves developing a graph of the mental steps taken by the solver in trying to solve a problem. This includes "back-tracking" steps where the solver backs up to a previous point in his solution after realizing that the path he is following will turn out to be fruitless. Hopefully this will give us further insight into additional ways in which IIDA might aid novice users in attacking a search problem. We have, however, just begun this analysis and have no results to report at this time.

1.5 Use of the Model

Having developed a general model for the search process (Figure 1), we then put the model into use. Consistent with our first goal for this segment of the project, we incorporated the model into the framework of the exercise mode of the IIDA program. Users of the exercise mode are taught to formulate strategy, to select and combine terms, to display a few items from the retrieved set and make a decision, based on the relevance of the records in the set, on whether and how to reformulate the search, and then either to reiterate the above steps or print out the complete set and log off.

The model also played a role in the development of the diagnostics IIDA will use in the exercise and assistance modes to determine whether a search is progressing well, and if not, what the nature of the difficulty might be. The "cycle-analysis" described below began in part by going to the individual search flow outlines to look for more specific indicators which could point to problems in reaching a successful conclusion to a search.

Each of the nine searches was examined in terms of some very simple, measurable parameters--such as the number of commands of the same type used in sequence, the number of commands in a cycle of select-combine type commands, the number of cycles in the entire search, and so on--to see if any of these parameters could be used to give an indication of how well the search was progressing. It was found that several of these measures could be used to detect poor searching behaviors. This information was then included in a set of "rules" which, when broken, will trigger a response from IIDA.

2. Diagnostic Procedures and a Search Exercise

2.1 Introduction

To review briefly the plan for instruction and assistance to IIDA users (4), IIDA will operate in either of two modes: exercise and assistance. In our original concept, there was a tutorial mode, intended to precede the other two in sequence of use. It would have provided basic instruction in searching. Since our revised goal is to prove the concepts of IIDA, rather than to operate a commercial service, we have omitted the tutorial mode, because of its expense, and will instead provide what basic instruction is needed to our experimental subjects by conventional techniques.

In exercise mode, IIDA reviews the basics of searching and gives the student user an opportunity to work on a simple search, using a limited subset of the full user language. Exercise mode contains three exercises. Exercise 1 reviews the basic DIALOG search commands (BEGIN, EXPAND, PAGE, SELECT, COMBINE and TYPE) and introduces the user to IIDA's HELP facilities. The user enters commands exactly as he is told to; he has no discretion. The purpose of the exercise is to show him the effects of use of commands.

Exercise 2 restricts the user to the same commands introduced in exercise 1, but he has more freedom in using them as he sees fit while carrying out a search assigned to him by IIDA. In doing this he will get experience in doing a brief, but actual, search and will begin to get familiar with IIDA's diagnostic procedures.

Exercise 3 introduces some of the more advanced commands and techniques. We again revert to a style in which the user is shown the effects of various usages, but there is no complete search to perform.

Assistance mode, which is, in effect, exercise 4, permits the user to perform any search whatever, in ERIC or NTIS, using any valid DIALOG commands. IIDA monitors progress and, when either invoked by the user or when it decides for itself that the user needs assistance, IIDA tells what problems it has detected and offers a variety of HELP services. The nature of the help available ranges from definitions of commands to advice on how to proceed, to an opportunity to begin again or review in exercise mode.

In our previous report (5) we provided a description of the first exercise. In this section we discuss the diagnostic procedures to be used in the second exercise and provide detailed specifications for that program.

It is our expectation that Exercises 1 and 2 will be completed and made operational by the end of November, 1978. Exercise 3 and the assistance mode programs will be completed by March, 1979. "Completed," in both cases, means ready for testing. The nature of the IIDA system is such that verifying that computer programs perform as specified is not sufficient. It is also necessary to try them out with users to verify that they were designed to do the right things. Further we expect there to be a number of adjustments in the values of control variables as a result of testing. Upon completion of tests with the complete system, we expect another round of revision before we declare the programs to be "ready."

Assistance mode is somewhat like Exercise 2, but differs in two important respects: the latter permits users to make use of only a limited number of DIALOG commands and the diagnostic procedures are less sophisticated. Diagnostic procedures for Exercise 2 have been specified in terms of 37 rules (an example of which is that if the user has created three consecutive null sets this fact is brought to his attention, both to give him some information about what is wrong and to indicate that something is wrong). We anticipate that assistance mode will require on the order of 100 similar rules.

2.2 Diagnostic Procedures

The procedures discussed below will be implemented in stages, and all require testing to verify their utility or to determine the appropriate level of various thresholds or parameters involved in their use. Some will be implemented during the ensuing quarter (October - December 1978) and some in the next quarter. Those planned for implementation prior to the end of 1978 are identified as such below.

2.2.1 Search Structure

Although work is being done, within this project, on a study of a general model of the search process, we do not ever expect to arrive at a point where searching becomes so routinized that, given an analysis of an incomplete record of a person's search, either IIDA or any human could prescribe exactly what is to be done to complete it successfully. We must continue, then, to deal with somewhat hazily defined measures and with heuristic procedures.

A search consists of a sequence of commands. Commands are classified, in IIDA, by type, according to a scheme depicted in Figure 5 and based upon one suggested by Penniman (6). For the most frequently used commands, the rationale is that BEGIN, END and LOGOFF commands all perform a similar function: they delimit the boundaries of a search. The commands EXPAND and SELECT retrieve information about individual descriptors or phrases. There are several variations on each command and they are subclassified as shown in the Figure. SELECT, of course, results in creation of a set based upon a descriptor, while EXPAND only provides information about that descriptor. COMBINE is a command that operates upon sets, not upon individual terms, hence it is a different type. It results in the creation of a new set by some Boolean combination of previously defined sets. TYPE and DISPLAY are virtually identical commands used to cause records or portions of the to be displayed on the user's terminal, a printer in the former case, a CRT in the latter. The purpose of issuing either command is assumed to be browsing: to look through a sample of records to determine whether a set has met the user's requirements. In some cases, one of these commands could terminate a search, but often, upon finding a likely looking set while browsing, the

c_type_maj	c_type_min	Command	comments
1	0	BEGIN	
1	1	.FILE	
1	2	END	
1	3	END/SAVE	
1	4	END/SDI	
1	5	LOGOFF	
1	8	LIMITALL	
2	0	EXPAND	yields a segment from alpha index
2	1	EXPAND	yields a segment from thesaurus
2	3	SELECT	single descriptor only
2	4	SELECT	single descriptor from E-table
2	5	SELECT	multiple descriptors from E-table
2	6	SELECT	contains an infix
2	7	SELECT	term is truncated
2	8	PAGE	used in context of EXPAND
3	0	COMBINE	all operators are "AND"
3	1	COMBINE	all operators are "OR"
3	2	COMBINE	mixed "AND" and "OR" operators
3	3	COMBINE	(same as above--distinction to be determine
3	5	LIMIT	

Figure 5. Classification of DIALOG commands into major and minor types.

c_type_maj	c_type_min	Command	comments
4	0	TYPE	with set argument
4	1	DISPLAY	with set argument
4	2	TYPE	with accession # argument
4	3	DISPLAY	with accession # argument
4	8	PAGE	used in context of TYPE/DISPLAY
5	0	PRINT	
5	1	PRINT	contains sort fields
5	2	PR-	cancel print command
6	0	EXPLAIN	
6	1	DISPLAY SETS	
7	0	.RECALL	
7	1	.EXECUTE	
7	2	.RELEASE	

Figure 5. Classification of DIALOG commands into major and minor types (Continued).

user wants an exhaustive print of the entire contents of the set and this is done with the off-line command PRINT, resulting in the computer listing being mailed to the user. These commands constitute command types 1, 2, 3, 4 and 5. Types 6 and 7 are infrequently used.

We define a string to be an uninterrupted sequence of commands of the same type. Thus, four consecutive COMBINE commands is a Type 3 string of length four. A sequence such as SELECT, EXPAND, SELECT, EXPAND, SELECT is a Type 2 string of length 5 -- all of the commands are of Type 2, even if they are different commands, just as END, LOGOFF or END, BEGIN are strings of Type 1, length 2, even though there are two different commands in each.

In virtually all of the searches we have examined, and also as reported by others (2, 6, 7), a search consists of a sequence of strings appearing in increasing numerical order as we have classified them. That is, a search might begin with a Type 1 string, followed by a Type 2, 3 and 4 in order. Then there might be another sequence of strings, beginning with a Type 1 or 2 string, and again proceeding to a Type 4 or 5. These cycles may continue over an extended period of time. We define a cycle to be a sequence of command strings such that, by our numbering system, the string type increases as the sequence proceeds. A string of a type lower than its predecessor begins a new cycle. For example, if a search consists of the following command types:

1. BEGIN
2. EXPAND
3. SELECT
4. SELECT
5. COMBINE
6. TYPE
7. SELECT
8. COMBINE
9. TYPE
10. PRINT

We consider this a two-cycle search. The first cycle starts with the BEGIN command (#1) and ends with the first TYPE command (#6). The next command (#7) is SELECT, which is of a lower type code than TYPE, hence the string and cycle both end and a new cycle begins with this SELECT command. As a generality, experienced searchers are parsimonious in terms of string length and number of cycles, but this alone seems not sufficient to discriminate between a well performed and a poorly performed search. We do, however, use these measures as indicators of other, more specific faults in a searcher's performance.

2.2.2 Syntactic Analysis

The complete logic of the IIDA syntactic analysis is presented in an earlier report (5). In the context of overall performance analysis, syntax analysis is done in order to detect specific mechanical errors which must be corrected. That, in fact, is the only meaningful definition of error in our search analysis: that an error is something which must be corrected, while other performance aberations do not necessarily have to be. A syntactic error is a fault in the composition of a command such that it will not be executed by DIALOG.

There is little that can be done upon discovery of a syntactic error other than to inform the user of the fact and request its remedy. IIDA can add information concerning the frequency with which this kind of error has been made by this user and it could, although it is not now so designed, impose information and exercises upon the user when he errs too much.

In IIDA, the detection and resolution of syntactic errors is done separately from detection and resolution of other faults, as shall be explained below.

2.2.3 Procedural Analysis

Once a command has been determined to be syntactically acceptable, we are concerned with its productive use and here we can only rarely say that a command is absolutely in error, and rarely can we tell a user exactly what to do about it. Hence, detection of procedural errors is all probabalistic and their remedy is all heuristic.

Procedural diagnostics are performed in the areas listed below. Those not being implemented for Exercise 2 are in parentheses and are marked with an *.

1. String and cycle statistics, used both alone and as indicators of other problems.
2. Repetitions of commands, both literal repetition and "essential" repetition, such as COMBINE 1 AND 2 and C 2 * 1, which have exactly the same effect.
3. Use of descriptors--checking whether descriptors whose use has generated null sets were checked in the thesaurus (and whether particular descriptors appear to be involved in an unproductive COMBINE string.*)
4. Sets created: number of null sets, unused (i.e., not referred to) sets, use of null sets in COMBINE or TYPE commands.
5. Thrashing, dwelling and convergence--these related concepts are defined in more detail below. They refer to repeated behavior in the formation of combinations of sets which leads to unproductive results.
6. Browsing -- (the searcher's behavior in selecting records to be viewed, possible repetitious selection and use of display formats.*)
7. Relevance -- the searcher is asked to make a relevance assessment of every record he has displayed to him, and these judgments are reviewed by IIDA to determine whether there seems to be progress toward attaining an acceptable set.

Thrashing, in general, is a pattern of rapid shifting of search direction on the part of the searcher and is probably indicative of his not taking the time to follow through on any one idea. Hence thrashing is considered to be unproductive. As a formal definition, thrashing requires a string of COMBINE commands of length L and an average value of the similarity index among the commands of less than S_1. L will be initially set at an arbitrary value, but must eventually be set by experience. The similarity index computes a measure of similarity of the set of descriptors used in COMBINE commands. S is computed as

$$S_{i,j} = \frac{1}{2}(d_{ij}/d_i + d_{ij}/d_j)$$

where : $S_{i,j}$ is the similarity index between two commands, i and j

d_{ij} is the number of descriptors in common to commands i and j

d_i is the number of descriptors used in command i

d_j is the number of descriptors used in command j

The computation of a similarity index ignores the boolean operators used, and is concerned exclusively with the descriptors used. The average similarity index value for a string of descriptors is the mean of the values of similarity between successive pairs of commands in the string.

Dwelling is a behavioral pattern opposite to thrashing. It represents a mode of use in which the searcher does not make significant changes in his searching patterns, but instead tries again and again to create a set which is only a minor variant of previously defined sets. Typically such a searcher is probably trying to refine a set beyond the sensitivity of the search language or data base to distinguish between similar definitions. Formally, dwelling is said to occur when a combine string exceeds length L and when the average similarity index value is greater than S_2, that is the requirement that the commands be similar.

When a person is dwelling, creating a set of closely related sets, it is also of interest to note whether he is, in fact, making any progress toward his stated goal. At the beginning of a search IIDA will ask a searcher to identify a goal, in broadly quantitative terms, e.g., a single good reference, a few good ones, or an exhaustive bibliography. This goal will be taken as a numeric goal by IIDA and the sizes of successively created sets will be compared with this goal to determine whether, on a purely numerical basis, the user seems to be nearing his goal. We identify five conditions:

--Set sizes are increasing toward the goal.
--Set sizes are decreasing toward the goal.
--Set sizes are increasing away from the goal.
--Set sizes are decreasing away from the goal.
--Indeterminate (i.e., the direction of movement is too erratic or there is no movement.)

When a searcher has been detected dwelling, this convergence information can be of extra help in pointing out to him what the effect has been.

2.2.4 Conversations with the User

Recalling that all but syntactic errors are not strictly classified as absolute errors, and that the IIDA programs cannot comprehend what is in the mind of the user, the objectives of the IIDA user conversations are: a) to describe problems detected by IIDA, b) to induce the user to confront them directly, and c) to provide hints on possible courses of action when possible. If these objectives are accomplished then we can reasonably expect the user to solve his own problems.

IIDA works under certain restrictions. It is necessarily reactive, i.e., IIDA can only react to steps taken by a user; it can not intuit what needs to be done and do it for him. All messages except those reporting absolute errors must be ignorable, i.e., they must be so worded that the user sees them as advisory in nature and knows that, if he feels secure in what he is doing, he may continue in the direction he has been going. The IIDA system must also be inoffensive. We do not want users to feel any conflict between themselves and the system or that the system is "behaving" in a haughty or patronizing manner. We want it to state potential problems in a straightforward, unemotional way. Finally, even where the user is making repeated errors, we do not want IIDA to have the appearance of nagging or badgering the user. Principally this last constraint means that at times we will suppress a fault-indicating message rather than be repetitious or boring.

A message control program has been developed within IIDA to meet the last of these requirements. Suggested by the TASK MONITOR of NLS-SCHOLAR (8). This program looks over the user's fault history and decides how to react to the totality of student performance, rather than just his last command. The others are largely met by the tone in which messages are written and by the decision never to cut a user entirely off the system unless his errors are such that he can not logically continue.

Whenever a fault threshold is exceeded, or a fault triggered, the appropriate diagnostic program reports this fact to a central Warning Control Program (WCP). It is possible that any given user command can trigger several warnings. The functions of the WCP are to decide to send messages, not send messages, step up or strengthen the severity of a message, or to add connecting phrases between multiple messages, as appropriate at any given time.

More specifically, the WCP is given a list of all faults triggered following any given command. It also has available the history of previously issued warnings. By scanning the current list of faults and recent previous ones, it can decide to:

1. Transmit a fault message as originally written.

2. Defer a message if the same fault has "recently" occurred and the user has been notified. The definition of "recently" will have to be determined experimentally, but, for example, we assume that if at command n he has been told he may have issued too many consecutive

SELECT commands, and if he issues one more such command at n + 1, there is no point rebuking him. He should be allowed to finish his thought and follow it through without being badgered. Whenever a message is deferred, a record is kept of that fact.

3. Suppress a message if another message covers the same fault but is more specific. For example, too many consecutive COMBINE commands is a general fault. Dwelling is a more specific fault of the same general type. Repeating commands is an even more specific fault. A history is not kept of suppressed messages, because the essence of the message would have been sent by another message.

4. Step up a message if a previously deferred fault has recurred. Thus, if command n exceeded a threshold for length of a string, and command n + 1 continued the string, we would defer a fault message at n + 1. If the problem continued, however, sooner or later it would be necessary to resume sending the deferred message and, when that happens, its strength should be enhanced by some phrase as "This has occurred m times since the last warning." How many times a message should be deferred is also to be determined experimentally.

Finally, a minor function of the WCP is to insert connecting words between fault messages. These, not yet designed at the time of this report, may be as simple as to add such phrases as "and also" between fault-describing messages.

If time permits, we hope to experiment with whether we can have the WCP respond differently to different patterns of user behavior. Thus certain kinds of faults might make the system more stringent, others less so.

Syntactic error messages are not subjected to the same kinds of analysis as procedural fault messages, because we feel that each syntactic error must be brought to the user's attention. We might use the WCP to add reminders when particular patterns of repeated error are detected.

2.3 Exercise 2

As indicated earlier, this exercise will familiarize the student with IIDA diagnostics. The student will be presented with a search problem and the suggestion of a general search strategy. Because suffixes and infix notations will not be used in this exercise, the statement of the search requirement will include terms which appear in the thesaurus.

The student will use the same commands used in Exercise 1 but he will be free to use them to create search strategies of his choosing, subject to the following restrictions: a) the PRINT command may not be used; b) a limit will be imposed on the number of citations listed using the TYPE command so that excessive listings may be avoided; and c) occasionally the student will be forced to call the HELP facility.

The progress of the search will be monitored by IIDA for syntactic and procedural problems. When a procedural problem is detected, the student is advised of it and may elect to use the HELP facility to ask for possible solutions to the problem. The student is free to accept or reject these solutions.

Use of the HELP facility in response to a message indicating a procedural problem, whether or not the recommended solutions are adopted, will begin to expose the student to the techniques of developing search strategy. Also, completion of this exercise will give the student experience in doing an actual search.

2.3.1 Program Description

The procedural diagnostics of Exercise 2 will be performed by a THRESHOLD ANALYZER, which consists of two PL-1 programs. The THRESHOLD ANALYZER compares various measures of student performance against assigned thresholds and generates appropriate messages if these thresholds are exceeded. These thresholds are written as a series of rules which reference the Student Data Structure (See Table 1) for information on the progress of the search, check this against the assigned values, and generate codes for messages which correspond to the surpassed thresholds. Message codes are processed by the WARNING CONTROL SUBROUTINE which establishes a priority for the issueance of messages within the context of a history of previously issued messages.

2.3.2 Threshold Analyzer

The two PL-1 programs which make up the THRESHOLD ANALYZER are called by IIDA as special actions. The first, or CANALYSIS program, executes those rules which require only a command from the student and information from the Student Data Structure as input; the second, or RANALYSIS program, executes those rules which require a command, data structure information, and a response from the host database. A sample of the PL-1 code for the THRESHOLD ANALYZER appears in Table 2.

The THRESHOLD ANALYZER rules are presented both in Table 3 and in narrative description form below. The threshold values cited are arbitrary, although rough hand simulations applying these rules to the transcripts of actual searches indicate that these values are useful as initial approximations. Subsequent testing will lead to substantial refinement of these thresholds.

Rule 1: If, after a successful log on with the BEGIN command, the BEGIN command is issued again, the student is advised to refrain from further use of this command and the current command is not passed to DIALOG.

Rule 2: If the PRINT command is issued, the student is advised that the command is illegal.

/* STUDENT DATA BASE: DATA ON THIS SEARCH */

```
command_history (100) ext,
   03 c_text char (50) varying,          /* argument of command */
   03 c_class fixed bin,                  /* 0=valid, 1=error, 2=control
   03 c_set fixed bin,                    /* set # created by sel & com
   03 c_type_maj fixed bin(7),            /* type_maj of command */
   03 c_type_min fixed bin(7),            /* type_min of command */
   03 c_string fixed bin,                 /* string number */
   03 c_group fixed bin,                  /* group number */
   03 c_time fixed bin (71),              /* time command was entered */
   03 c_time_diff fixed bin (71),         /* time since last command */
   03 c_stack fixed bin;                  /* stack number */

set_history (98) ext,
   03 s_text fixed bin,                   /* pointer to cmd history */
   03 s_size fixed bin,                   /* set size */.
   03 s_refs fixed bin,                   /* times referenced */
   03 s_rel fixed (3,2);                  /* average relevance of set */

error_history (100) ext,
   03 e_text fixed bin,                   /* pointer to command history
   03 error_type fixed bin,               /* error type code */
   03 ews (4) bit (1),                    /* error warning subroutine fl
   03 e_connect fixed bin;                /* # of connecting code */

sets_viewed_history (30) ext,
   03 r_text fixed bin,                   /* points to command history *
   03 set_viewed fixed bin,               /* set number */
   03 record_format fixed bin,            /* format of records viewed */
   03 record_range,
      04 first_rec fixed bin,             /* range of records viewed */
      04 last_rec fixed bin,
   03 rv_knt fixed bin,                   /* number of recs viewed */
   03 record_viewed (20),
      04 access_num char (8),             /* DIALOG acc # */
      04 relevance fixed bin,             /* user-assigned relevance */
   03 view_avg fixed (3,2);               /* average relevance for viewi

help_history (15) ext,
   03 help_text fixed bin,                /* pointer to HELP command */
   03 help_type (15) bit (1);             /* type of HELP called */

table_pointers ext,
   03 sdp (98) offset (sdb),              /* set descriptors pointers */
   03 dhp (100) offset (sdb);             /* descriptor histories pointe

set_desc based,

   03 s_desc_exp char (256) varying,      /* expanded descriptors */
   03 s_desc_norm char (256) varying,     /* normalized descriptors */
   03 s_tag_type fixed bin,               /* 0=none, 1=prefix, 2=suffix,
   03 s_lim_suf char (8),                 /* suffix on limit, if present
   03 s_d_num fixed bin,                  /* # of unique desc in set */
   03 s_desc (15) char (42) varying;      /* descriptors */
```

Table 1. Student Data Structure

```
descriptors_history based,
   03 desc_key char (40),                  /* key to the sdb record  */
   03 arg_usage (4),                       /* usage of argument */
      04 d_knt fixed bin,                  /* number of times used */
      04 d_cmds (25) fixed bin;            /* particular places used */


 area (32767) ext;                         /* area for set-desc and desc-

expand_data ext,

   03 e_table (50),
      04 e_term char (42) varying,         /* descriptor */
      04 e_items char (6),                 /* postings */
      04 e_rt char (3),                    /* related terms */

   03 r_table (50),                        /* as above, for relate */
      04 r_term char (42) varying,
      04 relationship char (1),
      04 r_items char (6),
      04 r_rt char (3),

   03 e_iida,
      04 t_time fixed bin (71),
      04 e_temp (50),                      /* as above, for IIDA issued [
         05 t_term char (42) varying,
         05 t_items char (6),
         05 t_rt char (3),

   03 expanded_idx (20),
      04 first_term char (42) varying,     /* first term seen in table */
      04 last_term char (42) varying;      /* last term seen in table */

support_data ext,

   03 index_data,
      04 c_last fixed bin,                 /* command hist */
      04 s_last fixed bin,                 /* set hist */
      04 e_last fixed bin,                 /* error hist */
      04 r_last fixed bin,                 /* records viewed hist */
      04 rv_last fixed bin,                /* last record viewed */
      04 h_last fixed bin,                 /* help usage history */
      04 d_last fixed bin,                 /* descriptor history */
      04 ex_last fixed bin,                /* EXPAND table entry */
      04 rel_last fixed bin,               /* RELATE table entry */
      04 ex_iida fixed bin,                /* IIDA issued EXPAND */
      04 ei_last fixed bin,                /* expanded index */
      04 stack_last fixed bin,             /* stacked command */
      04 v_last fixed bin,                 /* last valid command */
      04 g_last fixed bin,                 /* group count */
      04 st_last fixed bin,                /* string count */

   03 first_time_marks,
      04 s_first bit (1),                  /* sets created */
      04 r_first bit (1),                  /* records viewed */
      04 v_first bit (1),                  /* valid command */
      04 ex_first bit (1),                 /* EXPAND table */
      04 rel_first bit (1),                /* RELATE table */
      04 iida_first bit (1),               /* IIDA EXPAND table */
      04 e_r bit (1),                      /* 0=E-table, 1=R-table */
```

Table 1. Student Data Structure (Continued)

```
03 time_data,
    04 search_time fixed bin (71),          /* total search time */
    04 time_avg fixed bin (71),             /* average time btwn comds */

03 cit_total fixed bin,                     /* total number of citations v

03 cycle_data,
    04 group_data (10),
        05 group_start fixed bin,           /* pointer to first cmd in grou
        05 group_length fixed bin,          /* # of commands in group */
        05 string_data (10),
            06 string_type fixed bin,               /* type of string */
            06 string_start fixed bin,              /* pointer to first c
            06 string_length fixed bin,             /* # cmds in string
            06 dom_com fixed bin,           /* dominant type of command */
        05 group_rel fixed (3,2),           /* average relevance for group
    04 group_rel_hi fixed (3,2),

03 zero_data,                               /* zero set check */
    04 zero_knt_search fixed bin,           /* total # cmds zero postings
    04 zero_pct fixed (5,2),                /* percent zero commands */
    04 zero_knt_cons fixed bin,             /* cons zero commands */
    04 zero_knt_cycle fixed bin,            /* zero cmds in cycle */

03 error_data,                              /* total errors */
    04 error_total fixed bin,               /* total # errors */
    04 err_pct_total fixed (5,2),           /* pct total cmds in error */
    04 et_knt (500) fixed bin,              /* count of errors by type */

03 cmd_data,
    04 ct_knt (7) fixed bin,                /* total cmds by type */
    04 cmd_knt fixed bin,                   /* total valid cmds */
    04 set_knt fixed bin,                   /* number of sets in list belo
    04 set_list (15) fixed bin,             /* set#s isolated in COMBINE *

03 help_data,
    04 help_time fixed bin (71),            /* cummulative time in HELP */
    04 help_time_pct fixed (5,2),           /* pct of time in HELP */
    04 h_c_pct fixed (5,2),                 /* pct of commands calling HEL

03 rel_data,                                /* used to compute average rel
    04 rel_sum fixed bin,
    04 rel_knt_cycle fixed bin,
    04 rel_sum_cycle fixed bin,
    04 rel_knt_set fixed bin,
    04 rel_sum_set fixed bin,

03 rep_data,                                /* data about repetitions */
    04 rep_total fixed bin,                 /* total repetitions */
    04 rep_knt_cycle fixed bin,             /* count of repetitions in cyc
    04 rep_knt_search fixed bin,            /* consecutive repetitions */

03 set_ref_data,
    04 no_ref_total fixed bin,              /* total references to set */
    04 no_ref_cycle fixed bin;              /* total references in cycle *
```

Table 1. Student Data Structure (Continued)

```
        /* RULE 3 */
IF STRING_LENGTH(G_LAST,ST_LAST) >= 8 & C_TYPE(C_LAST) = 2 &
DOM_COM(G_LAST,ST_LAST) = 1 THEN DO;
E_LAST = E_LAST + 1;
ERROR_TEXT(E_LAST) = C_LAST;
ERROR_TYPE(E_LAST) = 403;
TE_SEND(3) = "1"B;
END;


        /* RULE 4 */
IF STRING_LENGTH(G_LAST,ST_LAST) >= 8 & C_TYPE(C_LAST) = 2 &
DOM_COM(G_LAST,ST_LAST) = 2 THEN DO;
E_LAST = E_LAST + 1;
ERROR_TEXT(E_LAST) = C_LAST;
ERROR_TYPE(E_LAST) = 404;
TE_SEND(4) = "1"B;
END;


        /* RULE 5 */
IF STRING_LENGTH(G_LAST,ST_LAST) >= 8 & C_TYPE(C_LAST) = 2 THEN DO;
E_LAST = E_LAST + 1;
ERROR_TEXT(E_LAST) = C_LAST;
ERROR_TYPE(E_LAST) = 405;
TE_SEND(5) = "1"B;
END;
```

Table 2. Sample of the PL-1 code for THRESHOLD ANALYZER.

CONDITIONS	1	2	3	4	5	6	7	8	9	10
	Rule Number									
string_length(g_last,st_last) ≥			8	8	8	5	5	10		
c_type(c_last)	1		2	2	2	3	4			
zero_knt_cons ≥									2	2
zero_knt_cycle ≥										
zero_knt_search ≥										
no_ref_cycle ≥										
no_ref_total ≥										
cy_change ≥										
rel_knt_cycle										
rel_knt_search										
num_ref_arg										
sim_avg										
view_avg(r_last)										
group_rel(g_last)										
group_rel_hi > group_rel(g_last)										
set_size_disp										
record_format(r_last)										
c_group(c_last) ≠ c_group(c_last - 1)										
time_avg										
cit_total										
dom_com		5	1	2					2	3
ACTIONS										
prepare message number	1	2	3	4	5	6	7	8	9	10
force call to help facility								x3		
record warning	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1
warning control program	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2
expand look-up subroutine									x3	x3

Table 3. THRESHOLD ANALYZER rules.

Rule Number

11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	
													4	4												
							4						3	3												
2																										
	3	3	3																							
				5	5	5																				
								3																		
									5																	
									0																	
										≥1	≥1															
										≤2	≥3	≥6														
							y																			
													≤t1	≤t2												
															≤t3			≥t5	≤t3							
																≤t4										
																	y									
																				c	d	s				
																			u							
																							≥t5			
																								f	g	h
	2	3		2	3																					
11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	
x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	x1	
x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	x2	
x3	x3	x3	x3	x3	x3	x3																				

y - yes; t1, t2, t3, t4, t5 - threshold values to be determined; u - uninformative format (to be determined); c - converging toward goal; d - diverging from goal; s - static relative to goal; f - greater than goal but less than 1.5 x goal; g - greater than 1.5 x goal but less than 2 x goal; h - greater than or equal to 2 x goal

Table 3. THRESHOLD ANALYZER rules (Continued).

Rule 3: If the EXPAND command occurs eight times consecutively, the student is warned.

Rule 4: If the SELECT command occurs eight times consecutively, the student is warned.

Rule 5: If Type 2 commands occur eight times consecutively, the student is warned.

Rule 6: If the COMBINE command occurs five times consecutively, the student is warned.

Rule 7: If the TYPE command occurs five times consecutively, the student is warned.

Rule 8: If the string length of any combination of commands is ten, the student is warned and forced to call the HELP facility.

Rule 9: If two consecutive null sets occur from the use of the SELECT command the student is warned.

Rule 10: If two consecutive null sets occur from the use of the COMBINE command, the student is warned.

Rule 11: If two consecutive null sets occur, the student is warned.

Rule 12: If three null sets occur in a cycle from the use of the SELECT command, the student is warned.

Rule 13: If three null sets occur in a cycle from the use of the COMBINE command, the student is warned.

Rule 14: If three null sets occur in a cycle, the student is warned.

Rule 15: If the total number of null sets thus far in the search resulting from use of the SELECT command is five, the student is warned.

Rule 16: If the total number of null sets thus far in the search resulting from the use of the COMBINE command is five, the student is warned.

Rule 17: If the total number of null sets thus far in the search is five, the student is warned.

Rule 18: If a null set is referenced in a TYPE command, the student is warned.

Rule 19: If three non-used sets are created during a cycle, the student is warned.

Rule 20: If the total number of non-used sets in the search thus far is five at the beginning of a cycle and this number is not reduced during the cycle, the student is warned.

Rule 21: If the number of repetitions (i.e., the number of occurrences - 1) of a command is two or less thus far in the search and at least one repetition occurs during this cycle, the student is warned.

Rule 22: If the number of repetitions of a command is three or more thus far in the search and at least one repetition occurs during this cycle, the student is warned and forced to call the HELP facility.

Rule 23: If the number of repetitions of a command is six or more thus far in the search, the student is warned and forced to call the HELP facility.

Rule 24: If, for at least four COMBINE commands using the AND or OR operators, the similarity index is less than the assigned threshold, the student is warned (thrashing).

Rule 25: If, for at least four COMBINE commands using the AND or OR operators, the similarity index is greater than the assigned threshold, the student is warned (dwelling).

Rule 26: If the average relevance (i.e., value of total relevance judgments/number of judgments) of documents viewed is less than the assigned threshold at this command, the student is warned.

Rule 27: If the average relevance of documents viewed this cycle is less than the assigned threshold, the student is warned and forced to call the HELP facility.

Rule 28: If the average relevance of a previous cycle is higher than the average relevance of this cycle, the student is warned.

Rule 29: If the average relevance at this command is higher than the assigned threshold the student is warned (the search may be complete).

Rule 30: If the average relevance at this command is less than the assigned threshold and the display format of this command is uninformative, the student is warned.

Rule 31: If the set size dispersion is converging towards the student's stated goal, the student is warned.

Rule 32: If the set size dispersion is diverging from the student's stated goal, the student is warned.

Rule 33: If the set size dispersion is static relative to the student's stated goal, the student is warned.

Rule 34: If the time between student commands is greater than the assigned threshold, the student is warned.

Rule 35: If the total citations listed thus far in the search is greater than the student's stated goal but less than 1.5 times the student's stated goal, the student is warned.

Rule 36: If the total citations listed thus far is greater than or equal to 1.5 times the student's stated goal, the student is warned.

Rule 37: If the total citations listed thus far is greater than or equal to twice the student's stated goal, the student is warned and logged off.

These rules which compose the THRESHOLD ANALYZER may be clustered by function into the following categories:

a. control for illegal commands (Rules 1 & 2)
b. control for consecutive commands of the same type (Rules 3 - 8)
c. control for the creation of null sets (Rules 9 - 18)
d. control for non-used sets (Rules 19 & 20)
e. control for repetition of the same command (Rules 21-23)
f. control for similarity of commands--thrashing and dwelling (Rules 24 & 25)
g. control for the relevance of documents (Rules 26 - 30)
h. control for the dispersion of set sizes relative to the stated goal for a final set size (Rules 31 - 33)
i. control for time delay between commands (Rule 34)
j. control for tital citations typed (Rules 35 -37)

When the application of these rules indicates a procedural error, this program references the Error Message Table (see Table 4) and turns on a bit to indicate that a message should be sent. The number of the rule in question serves as the index to the positions in the Error Message Table. The THRESHOLD ANALYZER also turns on bits in the table, when necessary, to signal: a) automatic log off; b) cancel current command; or c) force the student to call the HELP facility.

The bit configurations thus established by the Threshold Analyzer serve as input to the other main program component of Exercise 2, the Warning Control Program.

```
dcl 01 err_mess_table(50) ext,    /*error message table*/
  03 temporary_group,             /*good for current command only*/
    04 te_prefix fixed bin(4),    /*code for connecting message prefix*/
    04 te_suffix fixed bin(4),    /*code for connecting message suffix*/
    04 te_send bit(1),            /*turn on if message to go out*/
    04 te_help bit(1),            /*force user to call help facility*/
    04 te_logoff bit(1),          /*force logoff on user*/
    04 te_cancel bit(1),          /*cancel current command*/

  03 permanent_group,             /*good for entire search*/
    04 pe_last fixed bin,         /*no. of last command in which this msg. issued*/
    04 pe_knt fixed bin,          /*counts times the message issued*/
    04 pe_specific fixed bin,     /*number of more specific msg., if any*/
    04 pe_defer bit(1);           /*defer message*/
```

Table 4. Error Message Table

2.3.3 Warning Control Program

The Warning Control Program is a PL-1 program which is called by IIDA as a special action to establish priorities for the messages signaled by the Threshold Analyzer. The WCP communicates with the Threshold Analyzer by referencing a common data structure, the Error Message Table (see Table 4). This table has both a temporary and a permanent part so that some information may be compiled on a per-command basis and other data may accumulate throughout the search. The temporary section of this table is re-initialized by IIDA after each call to the Threshold Analyzer and the Warning Control Program.

For each threshold value which is surpassed during a search, the Threshold Analyzer turns on a bit in the Error Message Table (EMT) which corresponds in numbered position within the table to the number of the rule which was broken. For example, if rule 15 had been broken the Threshold Analyzer would turn on the te_send bit in the 15th position in the table. Since the rule number corresponds to the number of the message to be sent, turning on this bit signals the WCP that message number 15 is a candidate for transmission to the student.

The WCP scans the Error Message Table and for each candidate message signaled there the following rules are executed:

> Rule 1: If a given message has a more specific message associated with the same Threshold Analyzer rule, and the more specific message is signaled for output, then delete the send status of the generic message.
>
> Rule 2: If a message has been given recently (i.e., within the last five commands), the assign the defer status to it.
>
> Rule 3: If, on input, the status of the message is defer, then assign a code for the proper connecting message and assign the send status to the message.
>
> Rule 4: If a message has been issued often (more than five times), then assign a code for the proper connecting message and assign the send status to the message.
>
> Rule 5: If more than one message is scheduled for output, assign the code for the proper connecting message.

At the conclusion of this scanning, the WCP will have posted the final configuration of status codes to the Error Message Table. IIDA will then read this table and execute the actions indicated.

2.3.4 Expand Look-up Subroutine

At this time, this subroutine has not been completely designed. However, it will operate within Exercise 2.

III SEARCH PROCESS ASSESSMENT

1. Error Analysis

The error classification appended to this report (Appendix C) was empirically developed from analysis of over 50 transcripts of on-line searches. The development was a two-stage process. The classification categories were selected and defined in the first step. During the second step they were modified and refined using 48 "real" searches using a variety of data bases on three search systems.

Still, the classification should be considered a draft. It is not a trivial problem to develop a straightforward, consistent and precise classification of this type of error, and more work is needed before the proposed classification can be considered satisfactory. Although it will probably never be perfect, this analysis can shed light on the type and magnitude of errors made by searchers, and thus, provide information which could be used to decrease errors, either through user education or changes in system design.

Table 5 shows the results of the analysis of the 48 "real" searches. Fourteen percent of the commands contained errors which were transmitted to the system (excluded were error type I. A., errors corrected before transmission). The average number of errors per search was 3, with a range of from zero to 13.

Another count of errors was made from 40 transcripts generated during an experiment in which searchers performed a group of pre-selected ERIC searches on the Lockheed system (See Table 6). The searchers were divided into two groups, experienced and novice searchers. In this analysis all typographical and spelling errors (I. A. <u>and</u> I. B.) were excluded so the results are not strictly comparable to those shown in Table 1. This data shows that novices make twice as many substantive errors per command as experienced searchers.

The proposed classification is neither as complete nor as detailed as it could be. Since it was derived empirically, it contains specific categories for errors which <u>have been observed</u> to occur with some frequency, rather than specific categories for all possible errors. It can, however, serve to point the way to problem areas and to provide order-of-magnitude data. A more detailed classification which defines errors so specifically that a computer can recognize them, and which relates the errors to specific commands, could be developed from this classification.

A number of apparent causes of errors have been identified. In an order approximately paralleling the classification, they are:

1. Failure to type perfectly.
2. Failure to spell perfectly.
3. Failure to have mastered the command language.

Total Searches: 48

Total Errors not corrected before transmission: 147
(total Errors - I.A.)

Average Errors not corrected before transmission: 3
Range: Ø to 13

Total number of commands: 1034

Average number of commands: 22
Range: 3 to 72

Average errors/command: .14
Range: .04 to .63

Table 5. Error Analysis of "Real" Searches.

	Novices	Exp. Searchers
Total Searches:	24	16
Total non-typographical or spelling errors:	62	17
Average non typo or spelling errors: Range: Ø to 13	2.6	1.1
Total number of commands:	429	268
Average number of commands: Range: 2 to 33	17.8	16.7
Average errors/command: Range: Ø to .39	.15	.07

Table 6. Error Analysis of ERIC/Lockheed Experimental Searches

4. Failure to pay attention to results.
5. Failure to remember preceding commands.
6. Failure to understand the search logic.
7. Failure to understand the file structure.
8. Failure to use the controlled vocabulary correctly.

Some of these errors could be detected fairly easily by a computer monitor (e.g., syntactic errors). However, other analyses, such as distinguishing a misspelled term from a controlled-vocabulary term input incorrectly, may be done automatically only with great effort. Still other analyses depend upon the observer following, and making educated guesses about, the thought processes of the searcher--for example, deciding when the wrong logical operator was used.

2. Identification of Measures Which Discriminate Between Users

2.1 Introduction

There is a need for accepted and widely-applicable measures of searching performance. Presently, such measures do not exist. The goal of this research is to examine the feasibility of using the behavior of searchers in their communication with the machine as measures of performance. That is, it is the process of searching which is the focus of attention.

The attraction of using the process of searching, rather than the results of searches, to assess performance is that an important segment of the search process can be monitored automatically and unobtrusively by the computer. This is not true for search results. Nor is it true for manual reference searches for which the process is much more difficult to study.

In order to determine the behaviors which correlate with performance, i.e., skill in searching, it is necessary to compare searches which vary in success. One approach might be to look at the results of searches, and compare process to result; this is a part of the proposed research. However, since the measures of results of searches which are available--recall and precision--are only very rough approximations, this may not be the most productive approach.

A better way of selecting searches which vary in quality would be to first select searchers who vary in skill. Given that there is no objective way to select searchers by skill level, experience level will be used instead. The underlying assumption is, of course, that experience is strongly correlated with skill.

Thus the major research objective is the identification of the differences between the searches of users of online systems who have different amounts and types of experience. Searchers classified into several experience categories will be asked to search four search problems. Data will be collected on the background of the searchers and on over 20 process and outcome variables. It will then be possible to perform a variety of analyses which will contribute knowledge about the relationship between the search and factors which are believed to influence it, characteristics of the searcher. It should also be possible to relate the search process to the factors which it influences, the search results.

2.2 Objectives and Rationale

The primary objective of the proposed study is to identify those techniques which differentiate between the searches of persons with different overall amounts of experience.

This study also has several subobjectives. The first is to identify those techniques which differentiate between the searches of persons who are searching a database with which they are familiar and the searches of persons who are searching an unfamiliar database. The second is to identify the factors which contribute to success in searching. The third is to present descriptive information on errors made in searching. The fourth is to describe the utilization of various capabilities of the system.

Of the possible methodologies that could be used to accomplish the objectives, a quasi-experimental design has been selected because, in terms of economy and feasibility, it appears to be by far the best approach to the problem. In a quasi-experimental design one tries to simulate "pure" experimental design in a situation where one does not have the capacity to assign subjects randomly to treatment groups.

2.3 Methodology

Seventy-two searchers will each perform two of four pre-selected searches on the Lockheed/DIALOG system using ONTAP, the 1975 equivalent of the ERIC database. The searchers will be selected from five groups: novice searchers, moderately experienced searchers with ERIC experience, moderately experienced searchers with no ERIC experience, very experienced searchers with no ERIC experience.

Data will be collected on the background of the searchers. In addition over 20 process and outcome variables will be measured by examination of the search transcripts. Statistical techniques will be used to identify both the process variables which are the best discriminators between experience groups and the process and background variables that best predict the dependent outcome variables.

2.3.1 Subjects

The seventy-two subjects will be selected from searchers-in-general to conform to the characteristics of the five groups shown in Figure 6. The novice searchers will be randomly selected from the daytime Fall of 1978 Fundamentals of Library and Information Science (FUNLIS) class at Drexel University. The experienced searchers will be recruited from the community of working online searchers.

	DIALOG/No ERIC	DIALOG/ERIC
Very Experienced	12 (1)	12 (2)
Moderately Experienced	12 (3)	12 (4)
Novices	24 (5)	

Figure 6. Study Design

2.3.2 Variables

The variables relating to online searching can be divided into four types: a) environmental variables; b) searcher variables; c) process variables; and d) outcome variables. These variables are listed in Tables 7 through 10. The level of measurement is shown for the variables on which data will be collected.

The experimental procedure is designed to control for most of the environmental variables. The subjects will be given two of the same four searches to perform on the same database and the same search system. The requestor--the researcher--is the same for all the searches.

One can see from Table 8, the list of searcher variables judged to be important, that there are a large number and types of training and experience might affect online systems performance.

Data will be collected on all the process and outcome variables listed in Tables 9 and 10 except the need for help.

2.3.3 The Search Problems

The file to be used is ONTAP (File 201) on the Lockheed/DIALOG system. This is a static file which contains the 1975 ERIC (Educational Resources Information Center) file. It corresponds in all respects (data elements, searchable fields, etc.) to the regular DIALOG ERIC file for 1975 accessions. ONTAP contains about 35,000 references, approximately 12% of the ERIC file. The ONTAP file contains "answer sets" for 29 searches which were created by exhaustively searching the file. These answer sets have been equated to the results of a perfect search (100% recall and 100% relevance) for each of the 29 search topics.

The prepared ONTAP searches are categorized according to complexity: simple, medium and difficult. Four searches of medium difficulty which fit the following criteria were chosen for the experiment:

1. The topic is not technical.
2. The search is suitable for a wide variety of strategies.
3. It is simple enough for novices to handle, and difficult enough to offer some challenge for the very experienced searchers.
4. There are more than 3 documents in the answer set.

2.3.4 Experimental Procedure

Each searcher will conduct two searches. The novice searchers will perform the searches in the Drexel Information Science Laboratory by appointment. Since the experienced searchers will be scattered geographically, they will be recruited by telephone, and the data will be collected by mail. The experienced searchers will be given carefully

Variable	Data to be Collected On	To be Controlled For	Not To be Measured
1. Database			
a. Specific		X	
b. Cost		X	
c. Subject		X	
2. Search System		X	
3. The Search			
a. Characteristics of the requestor		X	
b. Objective of the search		X	
c. Complexity of the search		X	
d. Subject of the search		X	
e. The specific request		X	
4. Organization			
a. Type	N		
b. Management attitudes			X
c. Charging policy	N		
5. Physical space			X
6. Terminal		Partially	
7. System response time	O	Partially	
8. Machine-related problems (other)		Partially	
9. Access to search tools		Partially	
10. Presence of the researcher		Partially	

O = Ordinal level variable

Table 7. Environmental Variables

Variable	Data to be Collected On	To be Controlled For	Not To be Measured
I. Education			
A. Undergraduate			
1. Year of degree			X
2. Major field	N		
3. Minor field			X
B. Graduate			
1. Year of degree			X
2. Major field(s)	N		
C. Other			
1. Training in subject of database	N		
2. Training in mathematics or science	N		
3. Training in library science	N		
II. Online Bibliographic Search Training			
A. Years since training	I		
B. Type of initial training	N		
C. Continuing education	N		
III. Online Bibliographic Search Experience			
A. Total experience			
1. Number of searches ever performed	O	Partially	
2. Number of searches ever performed using a specific vendor system	O	Partially	
3. Number of searches ever performed on a specific database	O	Partially	
4. Number of searches ever performed on a specific database using a specific vendor system	O	Partially	

I = Interval level data
O = Ordinal level data
N = Nominal level data

Table 8. Searcher Variables

Variable	Data to be Collected On	To be Controlled For	Not To be Measured
B. Current activity level	I		
1. Number of searches performed per month	I	Partially	
2. Number of searches performed per month on a specific vendor system	I	Partially	
3. Number of searches performed per month on specific database	I	Partially	
4. Number of searches performed per month on a specific database using a specific vendor system	I	Partially	
IV. Other Experience			
A. Experience with reference searching	N		
B. Experience with hard copy equivalent of database	O		
C. Experience with computers	O		
D. Typing ability	O		
V. Personal Characteristics			
A. Intelligence			X
B. Creativity			X
C. Problem-solving ability			X
D. Cognitive style			X
E. Flexibility			X
F. Age			X
G. Sex			X
H. Attitude to online searching	I		
I. etc.			

I = Interval level data
O = Ordinal level data
N = Nominal level data

Table 8. Searcher Variables (Continued)

Variable	Data to be Collected On	To be Controlled For	Not To be Measured
*1. Commands used (by type of command)	I		
*2. Descriptors searched	I		
*3. Different types of descriptors used			
Thesaurus	N		
Free text	N		
Prefixes	N		
Suffixes	N		
*4. Errors by:			
Typographic	I		
Other (classified)	I		
5. Errors with potential impact on search results	I		
6. Errors with actual impact on search results	I		
*7. Use of sophisticated techniques			
Short logic form	N		
Stacking	N		
Truncation	N		
Adjacency	N		
Nested logic	N		
Printing in useful subsets	N		
*8. Number of records viewed	I		
*9. Number of sets viewed	I		
*10. String/cycle analysis	I		
*11. Search rating by knowledgable searchers	I		
*12. Requests for help			X

I= Interval level data
N= Nominal level data

*Computer monitorable
**Partly monitorable

Table 9. Process Variables

Variable	Data to be Collected On	To be Controlled For	Not To be Measured
1. References retrieved	I		
2. Recall	I		
3. Precision	I		
*4. Connect time	I		
5. Efficiency	I		
6. Searcher satisfaction			
Likert scale	O		
Semantic differential	O		
7. End user satisfaction			N/A

I = Interval level data
O = Ordinal level data

*Computer monitorable

Table 10. Outcome Variables

worked out directions which have been pilot-tested in advance.

Each of the four searches will be performed six times by the group of searchers in each "experience cell" in Table 6, and twelve times by the novice searchers. The searches will be randomized within each cell.

2.4 Results

A discussion of the results will be forthcoming.

2.5 Discussion

The proposed research is multifaceted. Although the main interest is in the differences between searchers at varied experience levels, the large body of empirical data collected in the study of the search process could be used for other purposes, particularly for designing monitors for online systems.

Information on the ways in which searches are actually being performed should be useful to system designers and educators as well. For example, a tabulation of errors made in searching might point out system features which cause special difficulty. Effort could then be made to correct the difficulty either by the system designer through changes in the system, or by educators through special attention during training.

It is expected that a major value of the proposed research is its potential contribution to the methodology of evaluating both the effects of searcher background on online system performance, and the systems side of the interface. This is an area where there is an acute need for work.

The major independent variables in this study are levels and types of experience. Experience was felt to be most suitable for this first effort because it is more likely to cause differences in searching behavior than some of the other variables. If differences in behavior due to experience are found to be measurable, the methods developed here can be refined and used to look for effects that may be more subtle. For example, there are a number of other important and related research problems having to do with the effect of subject knowledge, or of training, on search behavior and performance. From the systems side of the interface, there is the problem of evaluation of the effects of particular command languages.

IV REFERENCES

1. Newell, A., and Simon, H. A. Human problem solving. Englewood Cliffs, N.J.: Prentice-Hall, 1972.

2. Markey, K., and Atherton, P. ONTAP: Online training and practice manual for ERIC data base searchers. ERIC Clearinghouse on Information Resources, Syracuse University, June, 1978.

3. Morrow, D. I. A generalized flowchart for the use of ORBIT and other on-line interactive bibliographic search systems. Journal of the American Society for Information Science. 1976, 27, 57-60.

4. Meadow, C. T., et al. Individualized instruction for data access final design report. NSF Grant No. DSI 76-09737, Graduate School of Library Science, Drexel University, Philadelphia, 1976.

5. Individualized instruction for data access quarterly report number 1. NSF Grant No. DSI 77-26525, Drexel University School of Library and Information Science, Philadelphia, June, 1978.

6. Penniman, W. D. Rhythms of dialogue in human-computer conversation. The Ohio State University, Ph. D. Dissertation, 1975.

7. Standera, O. On-line retrieval systems: Some observations on the user/systems interface. Proceedings of the American Society for Information Science. 1975, 12, 38-40.

8. Grignetti, M. C., Hausmann, C., & Gould, L. An 'intelligent' on-line assistant and tutor -- NLS-SCHOLAR. Proceedings of the National Computer Conference, American Federation of Information Processing Societies, Montvale, N. J., 1975, 775-81.

APPENDIX A

On-line Searching Project
Instructions

As part of our research on on-line data base searching, we are trying to get a clearer picture of the procedures that are involved in conducting an on-line search. In order to more easily follow the way in which a searcher attacks a search problem, we are asking that you think out loud as you work through a sample search request that we will give you.

Specifically, we would like you to think out loud as soon as you begin looking over the search request, saying all the thoughts that come to you as you study the problem and begin to formulate your strategy for solving it. It will not be necessary to think out loud while actually doing the search on-line. Then after doing the search, we would like you to go over the transcript and again talk about what was going through your mind during the search, especially stating your reasons for conducting the search exactly as you did, including steps which did not lead to useful results, and whatever decisions you made while on-line, including any alternative strategies you considered but then rejected. We would like to emphasize that this is not a test of how well you search; we are primarily interested in how a search is generally conducted, and through what stages a searcher progresses in solving a request. Therefore, we would like you to be very specific about what you were thinking at each step of the search. Other than talking about your search procedure, however, we would like you to treat this problem as if it were one you received during your normal work situation, so that your search follows, as much as possible, the same procedure you would generally use.

We should also mention that your participation in this project will be kept confidential and that you will not be personally identified in any written or oral communication concerning this project.

APPENDIX B

SEARCH REQUESTS

Request #1

The user would like to find references containing gas chromatographic and/or mass spectrometric analyses of nitrosamine compounds (especially dimethyl- and diethylnitrosamine, but any others that can be found, as well).

Search the CHEMCON data base (File 3) on Lockheed and print out, on-line, the CA abstract numbers of all references retrieved.

Request #2

The user has found a paper of interest and would like to find references to all other work related to this paper. The paper is by David S. Auld in Biochem. Biophys. Res. Comm. (1976) and is entitled, "Yeast RNA polymerase I. A eukaryotic zinc metalloenzyme."

Search the CHEMCON data base (File 3) on Lockheed and print out, on-line, the CA abstract numbers of all references retrieved.

Request #3

The user would like some background material on catecholamines. He will be starting work with this group of compounds but knows little about them, so he would like to have just a few major references that can give him a quick overview of the state-of-the-art of this field.

Search the CHEMCON data base (File 3) on Lockheed and print out, on line, the CA abstract numbers of all references retrieved.

APPENDIX C

ONLINE SEARCHING ERROR CLASSIFICATION

DEFINITIONS

I. Typographical and spelling errors

A. Corrected before trainsmission

Any error corrected before transmission, either in the search term or in the command language.

B. Not corrected before transmission

An error in a command or descriptor that is not obviously a format or terminology error. When in doubt use VI.

II. Syntactic/semantic errors

A. Omitting commands

Forgetting to input a command code; i.e. selecting and combining terms as one would in the ORBIT system.

B. Combining descriptors rather than sets

In the COMBINE command, using full words instead of set numbers

C. Wrong command code

Code is valid, but is used in the wrong place.

D. Format errors

Incorrectly-formatted commands, or, commands in which the codes or punctuation conventions are incorrect.

E. Other

III. Procedural errors

A. Command unnecessary or repeated unnecessarily

Repeating the same command, or inputting a command that gives redundant information.

IV. Logic errors

A. Forming a set bound to produce zero postings

In a logical operation, combing terms in a manner so that the result is necessarily zero postings.

B. Wrong logical operator used

One logical operator (AND, OR, or NOT) substituted for another.

C. Failed to use already-combined sets

Re-creation, in a COMBINE statement, sets already created.

D. Wrong set number used

Creation of an unintended set, resulting from use of wrong set number(s); or, use of a non-existant set number.

E. Performed unnecessary logical operation

Logical operation is redundant; result should already be known from previous logical operations. (Prefer to III.A.)

F. Other

V. <u>Terminology errors</u>

A. Used incorrect subject term; correct term in thesaurus

Correct term could have been found using a cross-reference in the thesaurus.

B. Used incorrect subject form; correct form available in thesaurus

Used for cases when the spelling, punctuation, or ending is slightly different from term in thesaurus.

C. Used as a descriptor term not in thesaurus

Most frequently, this would be an invalid multi-word term which would receive zero postings.

D. Incorrect subject term format

Refers to mistakes in the adjacency features or labelling protocols for subject term descriptors. (Prefer to II.D.)

E. Non-subject term input incorrectly

Non-subject term input in the wrong format.

F. Term unnecessary; would be covered by another term

For example, searching for both a subject heading and a single-word descriptor that is part of the subject heading. (Prefer to III.A)

G. Other

VI. Impossible to classify/other

Used when in doubt or for all inexplicable entries.

INDIVIDUALIZED INSTRUCTION FOR DATA ACCESS
(IIDA)

Quarterly Report No. 3
December, 1978

Drexel University, School of Library
and Information Science
Franklin Institute Research Laboratories

NSF Grant No. DSI 77-26524

TABLE OF CONTENTS

I. OVERVIEW

II. EVALUATION

1. Formative Evaluation

   1.1 Project Staff
   1.2 Computer Science Majors
   1.3 "Real" Searches

2. IIDA as Assistant

   2.1 Fundamentals of Library and Information Science User Group
   2.2 Journeyman Searchers

3. IIDA as Instructor

   3.1 Technical Writing
   3.2 Business Writing
   3.3 Cognitive Style

4. Measures for Evaluation of IIDA

   4.1 Dependent Variables
   4.2 Independent Variables
   4.2.1 User Satisfaction
   4.2.2 Attitudes Toward Future User Behavior
   4.2.3 Problem-Solving Style
   4.2.4 Demographics

III. REFERENCES

I. OVERVIEW

This project is a renewal of earlier work on Individualized Instruction for Data Access (IIDA). Begun in July 1976, with initial funding for one year, the project was resumed in April 1978 and is to be completed in two years. This series of quarterly progress reports is planned in depth on selected aspects of the project and to contain a brief overall progress statement in each report.

The project staff are divided into two groups. The computer group is concerned with the design, implementation and testing of the requisite computer programs. From the user's standpoint there are four major subsections of the total system. In the first exercise the programs lead a user through a basic search in lock-step fashion, introducing some basic commands and providing familiarization with the general structure of a search.

The second exercise allows the user to do a constrained search. Although he is not free to use any search command at any time he is free to carry out the search pretty much as he wants. In this exercise a core set of diagnostic routines and rules are used by the program to monitor the activity of the user and provide various kinds of feedback or assistance.

The third exercise represents advanced search training in that, as in exercise one, the user is introduced to search commands and their use. The additional commands introduced here will include such things as variations on SELECT and the shorthand notations for DIALOG commands.

In the assistance mode (or "fourth exercise") the user is allowed to do an unconstrained search of his choosing. For this exercise the set of diagnostics and rules is to be expanded in order to deal with the considerably greater freedom which the user has relative to exercise two.

By the end of February 1979 the first and second exercises will have undergone both system de-bugging and preliminary formative evaluation by a small group of computer literate users looking actively for flaws in the system. With the second exercise providing the nucleus to be expanded into the assistance mode, both the third exercise and the assistance mode should be ready for use in evaluation testing in the spring of 1979.

The behavioral group of the project staff is concerned with both formative and summative evaluation of IIDA. In formative evaluation our concern is with monitoring system development and with providing feedback and information for refinement and further development of the system. For example, a number of the rules incorporated into the second exercise require the specification of a threshold value which, when exceeded by the user, results in the sending of a message to the user. These values are at present set by intuition or by arbitrary choice. Presumably use of the system will lead to revision of the threshold values.

In summative evaluation the concern of the behavioral group is with an assessment of the impact and effectiveness of the IIDA system and with the extent to which the objectives of the project are met. As indicated in the last quarterly progress report (1) the main topic for this report is the plans that have been made for summative evaluation.

In the body of the report that follows there is a discussion of four specific issues. The first of these deals with some aspects of formative evaluation planning which overlap with summative evaluation. The second and third have to do with specific plans for evaluation of the impact of the system on users. Given the structure of the IIDA system it is possible to ask two major kinds of questions. The first of these is about the effects of IIDA when the system operates only as an assistant. When dealing with this issue it is assumed that the user has previously had a reasonable amount of training in DIALOG searching and engages IIDA only through the assistance mode. The second major kind of question one can ask has to do with the effectiveness of the IIDA exercises in teaching new users how to do bibliographic information retrieval. When dealing with this issue it is assumed that the user has had no previous direct experience with searching and utilizes the capacities of IIDA as both instructor and assistant. The fourth and final major portion of this report is devoted to a discussion and analysis of the kinds of measures which can and should be used in assessing the impact of IIDA.

II. EVALUATION

1. Formative Evaluation

1.1 Project Staff

One type of formative evaluation of IIDA will begin with the availability of exercises one and two. Project staff members will become users of the system in order to have the experience of seeing what it is like and in order to look for flaws or ways to improve the operation of the system.

1.2 Computer Science Majors

Next a small group of undergraduate computer science majors will be recruited for the purpose of destructive testing. Because of the lock-step nature of exercise one, it is expected that the bulk of the destructive testing will be focused on exercise two.

On the Drexel campus there is an undergraduate organization called the Math and Computer Science Club. Contact with this organization has been made and several undergraduates have been recruited. These students are very enthusiastic about the opportunity to act as users and to push the system until its flaws show. In fact part of the initial briefing of these users will be to challenge them to find things wrong.

1.3 "Real" Searches

In addition, a number of searches done by real searchers will be re-done through exercise two in order to look at the responses of the system to "real" searches. The seventy-two searches to be done through exercise two will be taken from the study on search process assessment described in a previous report (1).

It is at this stage that the real core of the evaluation work begins. One component part of the search process assessment study, to be described in more depth in a subsequent progress report, is an attempt to establish "quality of search" scores for each of the searches collected. These scores will be derived from rating scale judgments of the quality of each search provided by "expert" judges.

This requires setting up criteria for what constitutes a good search, obtaining experienced searchers to make the judgments, and then having the experienced searchers rate each search on the criteria specified. If fairly high inter-rater agreement can be obtained then this would provide an independent measure of search quality.

Note that most of the diagnostic information kept by IIDA is, in effect, a set of measures of the search quality in that, for example, we would expect that the number of times particular messages have been sent to the user to be related to the quality of the search. If, then, the measure of search quality allows us to discriminate between the just trained users and the more experienced users who participated in the search process assessment study, the relationship between the search quality measure and the kinds of variables measured by the record keeping and diagnostic functions of IIDA can be explored.

Thus when the seventy-two searches are re-done through the IIDA second exercise we can accomplish two major tasks. The first is to explore the response of the second exercise to "real" searches while looking for flaws that need to be corrected. The second major task is the development of a set of criterion measures which can be used in subsequent summative evaluation work.

2. IIDA as Assistant

In looking at the issue of how well IIDA is able to perform as an assistant during the search we are basically concerned with users who have already had a reasonable degree of search training and consequently are to be exposed to IIDA only through the assistance mode. Ideally we should work with several kinds of user groups which differ in the amount of search experience and/or the kind of search training they have had.

The following studies are described in order of priority in that we do not seriously expect to be able to conduct all of the studies described in this report but we do expect to be able to complete several and the current list represents our priorities ranging from "must do" to "would be nice if." It should be also noted that the set of studies described below is shaped by consideration of the available resources for conducting possible studies. For example, the limited number of simultaneous or near simultaneous users that can be accommodated precludes doing certain kinds of studies. Given that users must be tested sequentially rather than simultaneously, any user group considered for this kind of testing must be one where we can have contact with various members of the group over a period of several days.

2.1 Fundamentals of Library and Information Science User Group

The School of Library and Information Science at Drexel University admits a number of new graduate students every year. When new students are admitted to the School they are required to register for a course entitled, "Fundamentals of Library and Information Science." One of the components of this course is a block of instruction in computer based bibliographic searching. This block of instruction, totaling roughly 12-13 hours, includes both classroom lectures and hands-on laboratory experience in searching. The Fundamentals students are recommended as a group for study not only by the fact that they are conveniently located for easy access, but also by the fact that they are similar to the intended IIDA user in that they do not have a great deal of search experience and can consequently be expected to run into difficulties. Hopefully IIDA will be responsive to these difficulties.

The study will be accomplished simply by adding on to the presently required search activities a further requirement. This requirement will consist of a standardized search request which will be the same for all students.

Since some of these students will ultimately go on to become intermediaries the statement of the problem will be much like those they can expect to receive in the future except that they will be unable to interact with the person submitting the request. Randomly, one-half of the group will be assigned to conduct the search with IIDA assistance while the other half will conduct the search without IIDA assistance.

2.2 Journeyman Searchers

Part of the search process assessment study referred to above and in an earlier report (1) involved recruiting active experienced searchers for study. These searchers were recurited from the Delaware valley region and are employed either as information retrieval specialists in private industry or in academic libraries. This group of searchers is the population which we intend to turn to for a study of the effects of IIDA assistance using the procedures already developed in the search process study and assuming that the earlier study has not exhausted the pool of people willing to participate.

The basic study design will involve two standardized search requests which are to be sent to each searcher who agrees to participate. Randomly, half of the searchers will be asked to do one of the searches first with the other half being asked to do the other first. Within each of these orders half of the searchers will be asked, randomly, to do the first search with IIDA assistance while the other half will be asked to do the first search without IIDA assistance.

While it might be simpler to conduct this study by asking each searcher to do only one search, either IIDA or non-IIDA, a differential return rate on the part of one group or the other would make the results of the study very difficult to interpret. With the design planned here there should not be a differential return rate and if there were it would seem to be very unlikely that the difference would be a function of anything directly related to differences bewteen IIDA and non-IIDA assisted searching.

3. IIDA as Instructor

In looking at the issue of how well IIDA is able to perform the service of enabling novice users to do a successful search we are basically concerned with users who have had no previous direct or instructional experience with computerized information retrieval. As mentioned in section 2, the studies described below are constrained by available resources and are described in order of priority. It should also be noted that we intend to do one study from section 2 and one from seciton 3 before doing any additional studies.

3.1 Technical Writing

Two years ago the Engineering College at Drexel University instituted a course requirement in Technical Writing for all engineering students. Each term there are several sections of this course offered. We have proposed to the faculty teaching this course that it would be a relevant experience for the students to learn something about bibliographic information retrieval. Many of the Drexel engineering graduates will ultimately be employed by organizations which utilize the services of information retrieval specialists. Presumably students who have had some direct exposure to searching should be better able to work with the people doing the searching.

The faculty involved with the Technical Writing courses have been very enthusiastic about the idea of incorporating IIDA instruction into their course and some have even offered us up to a week of in class time should we feel we need it.

For both experimental and pedagogical reasons each student will perform a search on a self-selected topic in each of two ways. One way in which the search will be done is through the mechanism of learning to do and actually doing a search with IIDA. The second way in which the search will be done is through the normal process of working with the intermediary in the library who will actually conduct the search. From the standpoint of instructional objectives each student will get a chance to learn something about the process of searching and about the process of interaction with an intermediary.

For experimental purposes half of the students will be randomly assigned to first doing the search through IIDA with the other half starting off with having the search done by the intermediary. This will allow comparison between the searches done by students with the searches done by a trained and experienced intermediary. The searches done by the intermediary provide a "bench mark" to be used in determining whether the students are able to complete a search with a reasonable degree of competence. It is also assumed that the areas of search performance where the students fall short of the standard set by the intermediary may provide us with some guidelines for improving the design of IIDA.

3.2 Business Writing

The same department responsible for teaching the Technical Writing courses mentioned above has also recently begun to offer at least two sections per term of a course in Business Writing. The course is designed to be for students from the College of Business what Technical Writing is for students from the College of Engineering. We have proposed the ideas outlined above to the faculty who have responsibility for the Business Writing course and have been met with considerable enthusiasm.

While a study done with students in Business Writing would be conducted in much the same manner as the one with students in Technical Writing we may want to conduct both simultaneously, treating them as a single study. While we would not have the option of randomly assigning students to curricula, this would allow us to look at the issue of whether students from different disciplines react differently to IIDA.

3.3 Cognitive Style

One of the conceptions that has guided the development of the whole IIDA system is that it is being designed for a scientifically oriented user group. It has been assumed that scientist and non-scientist types display different cognitive styles and that the scientist type of cognitive style will be more compatible with IIDA than the non-scientist type. Assuming that time and resources are sufficient, one of the things we should like to do is to administer a test of cognitive style to a relatively large number of students.

Study participants would then be recruited from this larger pool. Two groups would be formed from the extreme scores and one from scorers in the middle range. Comparisons of the performance of the various groups would provide information about differences in user reactions to IIDA as a funciton of cognitive style. Should major differences be found the information thus gained could be utilized in further design modification of IIDA in order to make the system more amenable to, or possibly more adaptable to, different types of users. One major unsolved issue for the conduct of this study is the selection of an appropriate test of cognitive style.

4. Measures for Evaluation of IIDA

The measures which should be useful in evaluation of the impact of the system are to be collected both through internal automatic record keeping functions of the system and through external means such as self-administered questionnaires, interviews, etc.

4.1 Dependent Variables

In formulating these measures it is important to keep the dependent variables in mind. In the case of IIDA as an assistant one is concentrating on

The following dependent variables:

a. Quality of the search - keeping in mind that we are interested not in the best possible search but in a sufficient search which satisfies the needs of the user.

1. Product quality - recall, precision, and relevance

2. Process quality

Note: much of this will be derived from the automatic record keeping of the machine.

3. Error rates - internal, external reliability

b. Efficiency

1. From a cost standpoint

2. Number of steps to get there

c. Reuse of diffusion - does the "user" intend to employ IIDA again and does he/she intend to encourage others to do so.

In the case of IIDA as a teaching instrument one is interested in these same variables as well as one which measures the rate of learning - given that the individual did not know how to search, how well is he/she currently doing. In other words, one would need some measures before IIDA was employed in the teaching mode.

In general, it will be important to asseble several data points on each of the dependent variables. This can be accomplished through the use of multiple searches. One is certainly interested in the rate of improvement over time - to what extent does quality improve or even the inclination to "diffuse" the innovation.

4.2 Independent Variables

Given the dependent variables described above, we will be looking at four classes of independent variables:

1. User satisfaction

2. Attitudes toward future user behavior

3. Problem solving or cognitive style

4. Demographics

4.2.1 User satisfaction

In the area of user satisfaction, we will be interested in the following type of question(s):

- The IIDA search(es) that I just completed was (were)

1	5
enjoyable	not enjoyable
very satisfactory	very unsatisfactory
helpful in working on a class assignment or problem	not very helpful
instrumental in working on an assignment/ problem	not instrumental
frustrating to use	not frustrating
stimulating to use	not stimulating
characterized by instructions that were easy to follow	not very easy

Note: in each of these cases of rating scale judgments one is tapping the attitudes or perceptions of the user. One has the choice of employing this at several points in time and measuring change or one can simply use it as a summative evaluation measure.

4.2.2 Attitudes Toward Future User Behavior

In the area of attitudes toward future user behavior, the following types of measures seem to make sense: (some of these also relate to problem solving style)

Agree strongly Agree Disagree Disagree strongly Don't know

- I do not like using the computer for classroom assignments
- My research is not enhanced through the use of a computer

- I would prefer to go to a librarian for bibliographic materials
- Intermediaries are preferable to a computer system
- Intermediaries are more comprehensive than I am able to be with the computer
- I would recommend that other people should learn bibliographic searching through IIDA
- IIDA is really limited to those with a background in the natural sciences
- IIDA is really limited to those with a background in computers

One would also want to ask the question before a search began, what are your expectations in using IIDA? Then, after the search was completed, one can ask, were these expectations met? Comparing before and after responses can be quite helpful in an evaluation of this kind. In this case, one might also consider a closed-ended question in terms of the expectations of an IIDA search:

- assessing whether bibliographic searching is useful to solve a particluar problem
- to learn to work with computers more readily
- to learn how to use this particular system

<u>4.2.3 Problem-Solving Style</u>

We view the area of problem-solving style as being one of the most interesting of the independent variables. We would start out by giving each respondent a description of the problem-solving process as we see it.

The problem solving process has the following stages:

- recognizing a problem
- defining the problem
- breaking it out into sub-problems
- selecting one of the sub-problems for "solution"
- generating options

- selecting an option
- implementation
- evaluation

Given this description, several questions seem important:

- at what point in the problem-solving process are bibliographic materials most useful
- at what points have you applied bibliographic searching
- at what points can you envision applying these resources
- where would you advise others to apply these resources

A different type of question attempting to measure the same dimension would read:

- when you have identified a problem, how do you identify the information resources that you will require:

  1. relying on colleagues
  2. relying on friends
  3. relying on a librarian
  4. relying on computer based bibliographic materials

4.2.4 Demographics.

Finally, we need to measure the following demographic variables

- age
- discipline/major
- degree
- courses in the sciences, social sciences, and humanities (how many and at what level)
- previous experience using bibliographic materials
- previous experience using computers
- employment background
- future plans

III. REFERENCES

1. Individualized Instruction for DATA Access. Quarterly Report Number 2. NSF Grant No. DSI 77-265525, Drexel University School of Library and Information Science, Philadelphia, September, 1978.