Tutorial Programs for Operation of On-Line Retrieval Systems*

D. E. CARUSO
Knowledge Availability Systems Center, University of Pittsburgh, Pittsburgh, Pa. 15213

Received September 22, 1969

The literature on interactive computerized search systems which make provision for the inexperienced user is summarized. Three such systems with varying user-training features, developed by the author at the University of Pittsburgh Chemical Information Center, are described. Novice-user interaction with these systems indicates that self-teaching systems can produce a competent user population which does not require an intermediary, human or mechanical, to create search strategies.

Early in 1968, the Chemical Information Center of the University of Pittsburgh was initiated. It was to serve students and faculty of the Oakland academic community and chemists in various research activities in the area.

Among the obligations assumed by the Chemical Information Center (CIC) at Pittsburgh are the development of new ways to introduce and use computer-based information services and the evaluation of the effectiveness of such procedures. We also assume the concomitant responsibilities of assessment of the effect of the innovative procedure on the subsequent use of the target service and, if possible, documentation of any changes in user-information-gathering behavior, which may result from use of the new procedures.

It was decided that interactive on-line computer programs, operating in the time-sharing environment, might provide one avenue to accomplish some of these obligations—that this environment might be a fertile one for devising new means of bringing the user into meaningful contact with and use of computer services. Accordingly, we set about assessing our resources and setting the parameters within which interactive on-line services to the local chemical community could be realistically devised and implemented.

Our first decision was to make our interactive systems maximally available to users. All information and directions for using the systems should be potentially available wherever the communications device, the typewriter console or display scope, was available. The display scope or the typewriter terminal should give access to all needed user instruction. Our next step was to review the literature for reports of any on-line interactive systems which incorporated self-teaching capabilities.

RECOGNIZING THE NEED FOR ON-LINE RETRIEVAL SYSTEMS

The need for some method of incorporating tutorial capabilities into on-line computer systems is becoming more evident as time-sharing systems proliferate. Remote access to computing capabilities implies use of those facilities by individuals who are intellectually as well as geographically removed from the computer. Data management and document retrieval systems seem to be among the first to react to this need, probably because the nature of the systems lend themselves to use by large numbers of different users, as a somewhat incidental part of their total professional responsibilities.

Recognition of this need was first expressed, tacitly, by the tendency to incorporate conversational prompting or guidance, or error diagnostics, into interactive fact or reference retrieval programs or by the various attempts to allow more or less free English in question expression to the system. The need is expressed explicitly by attempts to create frankly tutorial subprograms to be superimposed upon or incorporated into various on-line systems in operational use.

More and more closely the need is defined and stated in the literature. In the third issue of the *Annual Review of Information Science and Technology*, Silberman and Filep report: "another development that promises to become increasingly popular is the use of a built-in capability within various operational systems for the provision of instruction to operators in how to use that system."[1] They mention only two instances; however, the Morrill system[2] and Springer's description of the early System Development Corp. attempt to add CAI (Computer-Aided Instruction) to a real-time air defense system.

The report prepared for the National Advisory Commission on Libraries, *Technology and Libraries*, refers to the need for CAI programs to "bring some measure of efficiency to this process (learning how to deal with computer-based systems) and help avoid both unnecessary frustration from blind fumbling or excessive dependence on experienced library personnel. By 1975, it will probably be rare to find any significant degree of computer support without also finding a CAI element as an adjunct."[3] Moore School's Morris Rubinoff characterizes the need as "vital" if mechanized systems are to be "competitive in any way with standard library research technique."[4] My own feeling is that self-teaching systems, with capabilities of varying levels of self-teaching, to the extent that the user can modify the system effectively to suit himself, will be the key to more effective use of those systems than will be accomplished with any other teaching device; and most important, that this mode of instruction will

*Presented before the Division of Chemical Literature, 157th Meeting, ACS, Minneapolis, Minn., April 13, 1969. Work reported was supported, in part, by NIH Grant FR00250 and by NSF Grant GN738.

be the source of creative use of what will otherwise be a mechanically operated tool limited strictly to those uses devised by a small group of program designers, who are more or less removed from the ultimate users of the programmed systems.

We are not concerned here with attempts to create "every man his own programmer," but rather with the use of the large and complex programmed systems prepared and perhaps sold as packages for special purposes or for adaptation to special needs. We are concerned with efforts to allow the uninitiated user to approach and use, with little or no help, on-line document or data retrieval, computer-based, systems. Rather more work has been done than might be expected, considering the low level of development of operational full-scale retrieval systems. It is perhaps time to review the state of the art for more ready sharing of current results, to stimulate the bringing to light of unreported projects, and to suggest or stimulate additional efforts.

Wherever experience has been gained in the use of interactive on-line search programs by uninitiated users, there has been a strong interest in the development of methods to make those programs easier to use. A summary of interaction characteristics of eight document retrieval systems which made ease of use an important criteria is given in Figure 1. User input patterns fall into three categories:

> Those which allow the user to state his question in natural English, either with total freedom, or in a somewhat formalized statement, using a subset of natural English.
>
> Those which require that elements of a search request, usually search terms and logic, be identified and supplied to the system in a specified order, often in a rigid pattern and using logical operators. Varying levels of prompting, guidance, and feedback are provided.
>
> Those which use a special command language, usually very simple. Often the commands are executed by a single, mnemonically-labeled key.

The first approach is restricted by the difficulties of computer manipulation of natural language; it has been accomplished in very limited ways. The second method results in long dialogs which are expensive of user and terminal time. The user must identify the separate aspects of his information need and translate them into the vocabulary of the system, whether it be the indexing vocabulary or other sources of searchable arguments. Often he is required to create his own logical search prescription. While this procedure, in our present state of technology, allows more precise retrieval specification, it increases the importance of self-teaching devices when casual users are anticipated.

Neither of these approaches allows much freedom of search strategy creation, the free language input being somewhat more restrictive, in this respect, than the structured input or prompting systems. The inflexibility results largely from the user's inability to observe the logical capabilities of the search program and the inability of the programmed systems to interpret request statements adequately.

The command language systems make an overt demand that the user expend effort in learning something clearly unrelated to the subject he is studying, before he can begin to pursue his primary interest in the stored file. Whether the advantages of being in control of the search procedure will be viewed by the user as sufficient compensation for the extra effort is a complex question involving user and system characteristics with relevant user variables being subject to change precisely as a result of his interaction with the system.

These systems are basically variations of established systems as used in "batch" mode. They are, however, dedicated to the proposition that they can and will be used by the individual, who needs a question answered, himself. Those systems which seem to have had the greatest public use have the most highly developed instructional devices, and more and more attention is given to this aspect in later versions of programs which have been given any sort of "naive-user" exposure. A more detailed review of these systems has been given by Caruso[15] and others[5-16] cited in Figure 1.

DEVELOPMENT OF CIC INTERACTIVE SYSTEMS

Hardware Constraints. Our work is strictly delimited by the computing facilities available to us at the University. These are, however, extensive (Figure 2). Our interactive and semi-interactive programs make use of two IBM 360 Model 50 computers, which operate in the time-sharing service and in the batch-mode service, respectively. We use card, tape, or disk for input. The

	Input 1. Free English 2. Guided 3. Command language			Search Term Negotiation Feedback 1. Alphabetically near 2. Semantically near		Hit count 1. By term 2. Total for search		Logic 1. Sum 2. Product 3. Difference			Output 1. Citation 2. Associated index terms 3. Abstract		
	1	2	3	1	2	1	2	1	2	3	1	2	3
Audacious[5]			X	(If truncated)	(UDC)		X	X	X	X	X		
Bold[6, 7]			X	X	X	X	X	X	X	X	X	X	X
Dialog[8, 9]			X	X	X	X	X	X	X	X	X	X	
Easy English[10]	X						X	[X	X	X]	X	X	
Grins[11]	X							[X]			X		
Orbit[12]		X		(Uses printed index, thesaurus)		X	X	[X	X]		X	X	
Recon[13]			X	X	X	X	X	X	X	X	X	X	X
G.S.T.[14, 15, 16]		X		X		X	X	X	X	X	X	X	

[ ] = Supplied by program

Figure 1. Summary of interaction characteristics of eight document reference retrieval programs

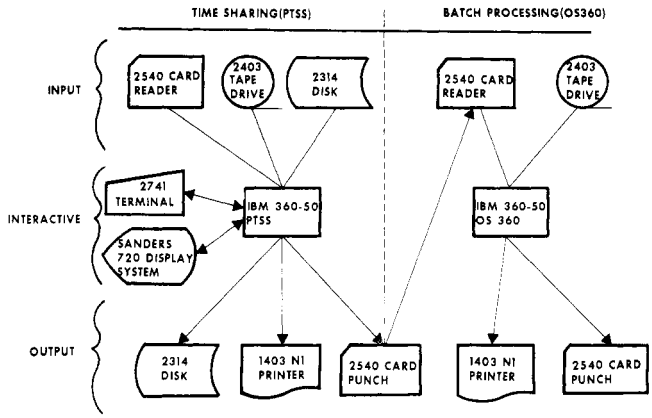

Figure 2. University of Pittsburgh computer facilities used in interactive systems development

IBM 2741 typewriter console and the Sanders 720 Visual Display Station are used interchangeably for interactive program execution, and disk and system printer are used as output devices. In some instances, output in card form is used as input for batch processing on the second computer.

The card reader and tape drive are important only at the initial input of data to our interactive systems. Storage for interactive programs must be on disk since tape involves delays in manual handling and in sequential accessing of records. The use of disk only as the storage medium has not been a hardship as disk maintenance is highly reliable and is well protected by backup and failsafe mechanisms, and disk storage space is available in excess of current requirements. The 2741 terminal prints slowly and is not suitable for large amounts of printed output. The Sanders 720 Visual Display Station is quiet and displays an adequate amount of text, but there is no hard copy of the interactive exchange.

The 1403 line printer is used for program and dataset listings. The 2540 card punch is used to prepare output from some interactive programs, to be used as input to the batch mode programs operating on the second IBM 360/50.

We have restricted our use of core-storage to 32K; larger amounts are available (up to 131K per user), but use of large amounts of core frequently degrades service and response times.

Software Constraints. As used here, software constraints include capabilities of the computer system and of personnel. We have sacrificed program efficiency to save programming time; all programs are written in high level languages: PIL (Pittsburgh Interpretive Language) or in CATALYST, a newly developed CAI (Computer-Aided Instruction) language. Programs can be revised easily and quickly, even during execution.

Constraints Imposed by the Services Exploited. No attempt was to be made to develop new services for the chemical community. Our efforts exploit those services presently available for purchase, rent, or subscription. The results of our work may suggest modification of existing services or indicate where new services may be needed.

The types of files to be exploited are here categorized by the characteristics which impose strongest constraints on use in interactive systems:

Cumulating files of data for fact retrieval, such as the Sadtler Research Laboratories Infrared Spectra

Recurring and superseding files of document citations for current awareness services, such as the *Condensates* tapes of Chemical Abstracts Services

Cumulating files of document citations for retrospective searches, perhaps a cumulation or compilation from the *Condensates* files.

Each of these types of files poses different problems when interactive systems are to be implemented. The differences arise primarily from:

Practical possibilities of file organization for fast access and search, a function of file content and updating

Kinds of use anticipated for each file

From the tutorial viewpoint, the most important use characteristic may well be the number of personal accesses which can be expected for a given service by any one user.

The Plan of Action. An ultimate goal of on-line interactive systems would include universality of information sources and total freedom of query mode. Something less is presently possible. In an effort to determine where our first effort could make the greatest contribution toward the ideal or ultimate goal, we reexamined present methods of utilizing the files we plan to exploit. For each file or service, there exists a currently operational batch mode search system.

In each of the operational batch mode systems there are major elements: the user, the question, the file, the search, and the output. Procedures for handling each element are more or less established for batch searches.

For interactive on-line searching it is obvious that file searching programs, file organization, and methods of output need revision. Response time as well as efficiency of searching also become more important in interactive services.

Focus of CIC Interactive Systems Development. As we attempt to make the service interactive, with the user in direct control of the process, another class of problems arises. The question-analyst or information specialist, who, knowing the system and its contents and capabilities, questioned the questioner and translated his needs into a problem which the system could answer, is gone. The system, by instructing the user, and the user, by his own reasoning, must take over this function. This area of user-system interaction was selected as the focal point of our efforts.

Interactive programs require a tutorial approach. We have noted earlier a growing emphasis on techniques for teaching the new or infrequent user how to operate the system. We decided to take a frankly tutorial approach to the problem of user-system interaction, utilizing as appropriate, the current understanding of computer-aided instructional techniques—particularly those techniques which utilize simulation of the actual system to be taught.

To present the user with a realistic situation in which to learn and to operate a retrieval system, it was necessary in each instance to create a whole retrieval system or simulate a complete system. The general design of such a system, derived from current practice, is simple. A program is written which operates on a file of data, to allow a user to state an information need, to report to the user on the closeness with which the system can

match answers to his request, and to permit the user to restate his needs, if possible, in terms which may more closely relate to the content of the file being searched. The elements of the interchange alternate until the user or the system decides that a satisfactory question statement has been achieved, whereupon the search is executed, and the results of the search are reported to the user.

Our own earlier experience in developing and testing a primitive on-line retrieval system for independent user operation indicated that the major problem areas of the interaction could be sorted into six areas:

1. Getting the user on-line and into the programmed retrieval system—the mechanics of console operation.
2. Getting the user's query into "system-acceptable" terminology and into a valid and processable strategy statement.
3. Performing a rapid search of the file to determine the number of documents which match the inquiry parameters.
4. Providing meaningful and user-understandable preliminary reports on search results.
5. Procedures for possible revision of the search prescription, with return to point 3, above.
6. Final output of search results.

Of the listed items, only 3 and 6 are not integral parts of our designated area of investigation. The search is necessary, however, to provide the user with information to assess his search strategy and to revise it. The final output can verify his judgment.

We have stated that our own efforts lead to the achievable goal of fully interactive search of complete files of given and definable content: data files, current awareness tapes, or a complete bibliographic record for retrospective search. We are presenting the files in this particular order, because they seem to present successively more difficult problems in the development of completely interactive searches: the data file, relatively small, highly organized, and infrequently updated; the current awareness file, again relatively small, but frequently superseding itself; and the retrospective file, where file size and updating are both problem areas. Despite the varying complexity of the file organization and maintenance problem, all present a similar pattern of sensitive user-system interaction areas. We have made a beginning, for each category of file, directed to the intermediate but distant goal of fully interactive searching of that file type. The nature of the beginning varies, in each instance, as we have tried to make the best application of currently available computer capabilities to the particular uses and difficulties of each type of file organization.

Development of the systems has not proceeded in that order, however, nor is development continuing evenly for each system. The most difficult, the retrospective search, was tackled first; by simulating the large-scale search of the huge file, we allowed more time and effort to be devoted to the tutorial aspects of the search strategy common to all search programs.

The second program to be developed, the *Condensates* tutorial, shows more effort expended on the problems of user interaction with large files using uncontrolled, unstructured indexes.

The third program, least developed, the infrared data files, emphasizes the aspect of controlling and searching the entire data base.

Immediate goals, those capable of realization with present resources of personnel, hardware, and software, have been set for each type of file. These realizable goals are summarized below. Details of each system are given in the next section on the status of projects.

For the cumulating file of data: The semi-interactive, predictive search, which operates to a certain point on time-sharing; then automatically transfers to a queue which is submitted to the OS system for batch processing at stated intervals. This results in a completely processed information request.

For the current awareness service, recurrent and superseding files: A brief tutorial which teaches how to create profiles for a specific current-awareness service. It does not presently search a complete file because of limited high-speed storage capacities (i.e., core) and because we have not attempted, as yet, large-scale operational inversion of these recurrent files, which for their intended use, supersede earlier issues at regular, frequent intervals.

For the cumulating file of document citations, the bibliographic record for retrospective searches: A tutorial for general principles of question analysis and logical structuring of questions. This program makes no attempt to relate to any particular file content and has its own data base. This base is easily changed to suit groups of different subject backgrounds, however. Its intended user-group—anyone who frames a question for a retrospective search.

Status of the Projects. Steps in the development of an interactive system may be isolated: (1) conception, (2) design of system, (3) preparation of suitable data base, (4) programming and coding the system, (5) debugging the program, (6) preliminary use tests, (7) revision of program design, recoding, and debugging, (8) controlled testing of systems with samples of the intended user population, (9) evaluation of completed system. For each of our defined three types of files, an interactive system has been developed and completed to varying degrees. Only the general strategy tutorial with searches through a portion of a bibliographic file has been completed through step 9. Progress is summarized for each system in Figure 3.

The remainder of this section of the paper will detail progress on each system. In each instance we will describe the specific objectives of the program; the files used, including any modification of the file, the design and operation of the program; any experience with use of, and present status of the program; and projected further development of the particular system.

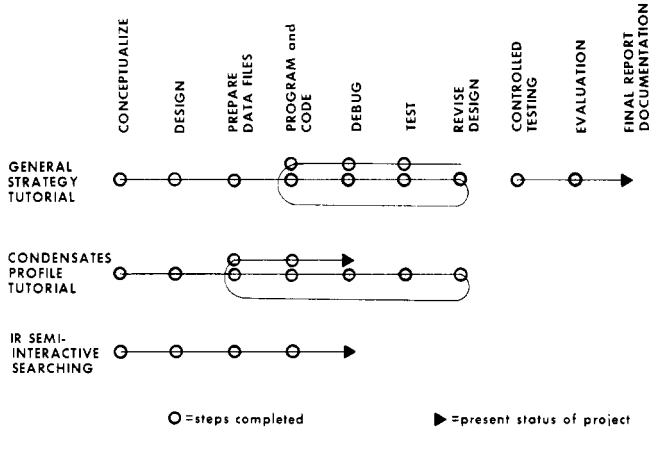

Figure 3. Status of interactive systems projects
June 30, 1969

Cumulating Files of Data. SPECIFIC OBJECTIVES. The objective in developing an interactive service for this file was to provide as effective a working tool as possible, within the hardware and software constraints noted, for actual day-to-day use by the working chemist, researcher, or student. The entire file was to be made available for immediate interactive search, on-line, with immediate listing via the console, of search results.

FILE UTILIZED. The Sadtler Research Laboratories search tapes of infrared spectral data are used as the basic file for this program. The data are provided in a sequential file by Sadtler; all information for each spectrum is grouped together. For more rapid searching, the data are rearranged in series of inverted files.

DESIGN AND OPERATION OF THE PROGRAM. Preliminary searches, using the batch mode search program supplied with the data files, indicated that the inexperienced user was likely to generate great amounts of output, confusing and useless to him and tying up the typewriter console for inordinately long periods of time.

It was decided that a preliminary, predictive search be developed, using a 2% selection of the file, which could be repeated as necessary, before being applied to the total file. As presently developed, this preliminary search, when approved by the chemist-user, gives some preliminary results via the console. The search prescription, as he has developed it, is then stored for a batch run on the Operating System 360 using the supplier's search program. The batch run is automatically formatted and a complete deck, including appropriate job control cards, is the final product of the interactive program.

CURRENT STATUS OF THE PROGRAM. The steps of the interactive search as presently implemented are (see also Figure 4):

1. Requests the parameters of search from chemist-user.
2. Checks parameters for appropriateness—e.g., if chemical class is specified, does it conform to those classes defined by Sadtler in its index?
3. Performs preliminary predictive search if desired; reports results.
4. Allows options at end of predictive search to: (*a*) complete search of full tape, (*b*) revise search, and (*c*) terminate.
5. Adds unique identifying case number to chemist-user's search statement and formats his statement to be accepted by the assembly language search program provided by Sadtler.
6. Reports the case number to chemist-user.
7. Adds the search request to others to create a batch for searching off-line on the 360 O.S.

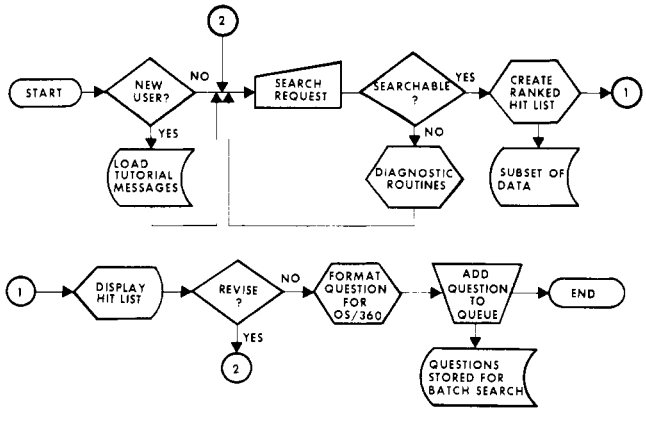

Figure 4. Semi-interactive search of IR data files

Results of the batch processed searches are transmitted to the Chemistry Library where copies of the spectra identified as relevant for each user's question are made from the microfilm files and are then delivered to the individual user. Copies of typical interactions and of a batch of questions prepared for presentation to the System 360 Operating System Assembly Language Program are attached as Appendices II and III of our progress report dated March 31, 1969.[14]

At present it is necessary to create a card deck of the search questions for presentation to the O.S. 360. Methods for sharing access to the 2361 disk by the two resident computers, one of which operates in the time-sharing mode and the other handling batch processing, are being developed by the University Computer Center and will allow direct manipulation of the batch search via the 2741 console in the near future.

Note that the preliminary search (item 3, above) is optional. In effect, this creates a kind of special purpose remote job entry station, which performs all the tedious details of coding the query, formatting and punching the complete question deck, including job control cards.

Also note that item 4,*c* allows the user to terminate the search. He may exercise this option because he has retrieved sufficient information and does not need a more complete search, or because he has determined that there is not likely to be data in the file which he can use.

NEXT STEPS IN DEVELOPMENT OF THE IR SEARCH PROGRAM. The system reported here represents the most preliminary efforts. Next steps in its development, in order of undertaking should be:

1. Extend the data base for the predictive search to include all relevant data for the 2% of the spectra included.
2. Test the system, using the present interactive-to-batch program, in actual user-environment as predictive search (which may suffice for some users) and as a remote-job entry station for the batch mode service.
3. Complete the conversion of the entire data base to inverted files which may be searched interactively, on-line, either directly or after the use of the predictive search.

The Profile Development Tutorial. SPECIFIC OBJECTIVES. The program has this objective, in terms of exit behavior: The individual who has completed the program will be able to develop a profile of his own subject interests which will conform to the logical requirements of the search program and locate in the biweekly issues of *Condensates* all or most of the articles which he would be interested in, while excluding those of little interest.

While the chemist-user may not be interested in the formal logics used, he does need to know the idiosyncrasies of the search program which affect the value of his profile. He needs to recognize the problems of searching uncontrolled, keyword indexes, and matching author terminology. The possible misuse of truncation must be understood, as well as its value in helping to provide matches for variant forms. The parameters of the profile must follow the prescribed logical form.

We hope to develop these capabilities by allowing the individual to experiment with the logical principles and

techniques of profile development in the environment in which the profile is to be used, i.e., in actually executing a computer search of the file. Finished profiles, except for revision, will be run thereafter in batches for the current awareness function.

FILES USED. The files used are those tape records issued by Chemical Abstracts Service as *Condensates*. They consist of references and keywords (searchable, word-by-word) in a sequential file. Each record is associated with the *Chemical Abstracts* abstract number. For use in the tutorial program, a selection of 120 references, arbitrarily chosen from one tape and dealing with polymers primarily, is used. The references are maintained in the sequential file. An alphabetically ordered, inverted file of searchable arguments was created in a very simple form:

Term	Abstract No.
POLYMERIZATION	0693783

DESIGN AND OPERATION. The *Condensates Profile Development Program* follows a simple teaching logic. It develops concepts and allows practice in their application. The tutorial program incorporates search routines and files of documents, which allow the user to practice what he learns and to experiment with variations of search term and strategy prescriptions. It is directed to the professional chemist whose own practical objective in using it is simply to learn how to state his information needs to a current awareness service.

He is provided a general type of strategy: logical conjunction of disjunctive clauses, with optional use of a negative clause. No time is wasted on explication of any other possible strategy, as most useful types of search logic are possible within the permitted framework.

For the chemist to run the tutorial program independently, he must, at the very least, know how to get the program running and have a general knowledge of its purpose. A minimal set of instructions has been prepared for his use (Figure 5). Once the program is running, there are stored 'helps' which he may call out to answer program-related questions, by typing 'help' in lieu of any other response.

A more extensive reference manual[17] has also been prepared. It covers details of mechanical operation and further develops concepts introduced by the computer program.

After a brief introductory tutorial which gives instruction on term selection and logic, the user enters the main section of the tutorial, the iterative searching program. Figure 6 shows these major user-system interaction points:

1. User identifies himself and is given a brief summary of the subject content of the stored file.
2. He lists terms to be used in building a profile statement.
3. The system tells him which terms are not found in the file and the number of times 'hit' terms were found.
4. For any terms which were not truncated, the system displays a list of alphabetically near terms. If truncation was used, it is assumed that he has already retrieved references on all desirable neighbor terms.
5. User may add to his term input if he wishes.
6. User arranges his terms into logical parameters.
7. Search is executed and the system reports total number of references retrieved.
8. User may ask to see as many of the references as desired.
9. The profile just executed is put into sentence form and presented, for comparison, with the output it produced.
10. User has the option of rearranging his profile, adding new search terms if desired. If revision is chosen, he reenters the program at step 5.

The *Condensates* Tutorial consists of (1) a small file of document citations from one issue of *Condensates* and (2) an interactive retrieval program which operates on that file of documents. The interactive retrieval program uses the same logic for searching the small document file as does the batch mode program which searches the complete *Condensates* tape to produce the biweekly alerting service.

In producing a good profile, or search prescription, the object is (1) to find the documents of interest to you but (2) to exclude all others. The tutorial program takes you, step-by-step, through the procedure used to create and execute a profile. The end-product of the program is a list of the citations it retrieves from the file, using the profile you create.

Since the document file is so small, you are not likely to find your real interests reflected in it. Try to compose a simple interest profile relative to the subject content of the file. The program will tell you what the current file consists of. To begin:

1. Turn the console on.
2. Hit the ATTN (attention) key (upper right of keyboard).
3. Type the following message (exactly) on the keyboard:
   $$ logon xxxxxx*, size=32000.
4. Hit the RETURN key. This sends the message to the computer.
5. The computer will respond:
   M:ACCEPTED
   Ready:
   If it does not so respond, repeat steps 2, 3, 4.
6. Now type the following message to start the first section (tutorial):
   load dataset tutor.
7. Hit the RETURN key. (You must hit the RETURN after every line you type.)
8. You will be asked to type your name.
9. The computer will type the following message:
   PROGRAM:SECTION=
10. Wait for the computer to provide further instructions.
11. If system failure occurs, or an ERROR condition occurs, try typing:
    Go.
    If the program does not resume operation, type:
    $$ logoff.
    and begin again.

* xxxxxx - You will be issued an identification number to be used here.

Figure 5. Minimal user instruction sheet for profile tutorial

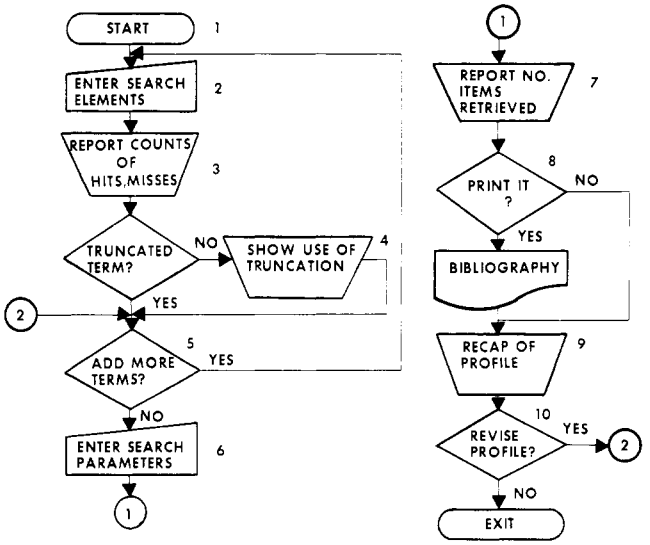

Figure 6. User system interaction profile development tutorial program

The program performs a binary search on the inverted file, locating exact matches for untruncated terms and matches for the indicated stems. The documents found to match the strategy requirements are located in the sequential file and listed if requested by the user.

EXPERIENCE IN USING THE PROFILE TUTORIAL. An earlier version of the profile tutorial required that the user choose one of a given list of documents from the stored file as a target and try to create a profile which would retrieve that item. The preliminary concept-developing tutorial had not yet been inserted—only the search routines as shown in Figure 6 were available. Twelve graduate students from the Department of Chemistry used that version of the program.

These 12 students had no experience with computer use. The tutorial was their first introduction to computer capabilities. Their use of the search program was completely independent—no aid was available to them. They used the minimum instruction sheet shown in Figure 5. The program contained the instructions for choosing a target-document.

System failure prevented program completion in two instances. Seven individuals were able to isolate their chosen target documents using the search program. One of the three failures resulted from misuse of the negative parameter. The two remaining failures were the result of poor parameter structure. In all instances the terms chosen for searching were adequate for the successful resolution of the problem.

Complete understanding of the use of the search logic—i.e., creating logical products of logical sums via appropriate parameter construction—was not demonstrated. The small size of the file contributed to the level of success that was achieved.

As a result of the experience with these students, a brief introductory session of the programmed instruction type was prepared to clarify the purpose of the experience and to provide guided practice in the specification of search terms and the use of the logical statement to expand or narrow the search area. Concomitantly, some of the tutorial features of the search program were eliminated as now redundant, and the size of the document and index files were increased to 120 and 960 items, respectively.

PROJECTED NEXT STEPS. Preliminary testing of the new tutorial to determine whether vocabulary and instructional format are clear to the chemist-user has been completed. A planned and controlled use test of the tutorial program with a sample of new entrants to the CIC sponsored *Condensates* service will be made. The test will use the channels already established by the CIC behavioral study group for feedback from the user-group.

We will look for any detectable differences between the groups of users who enter the service via the established seminar and individual counseling path and any who enter via the interactive tutorial-practice program.

The General Strategy Tutorial. SPECIFIC OBJECTIVES. The General Strategy Tutorial Program is an ambitious one, which attempts to build an understanding of general principles of file searching which can be transferred to any search situation. A controlled experimental study of its use is the subject of a doctoral dissertation,[15] and a description of the program with illustrations of its use are included in the *Quarterly Report* of the CIC, July–Sept. 1968.[16] Only a brief summary will be attempted here.

The isolation of appropriate subject aspects from a natural language question, translation of those aspects into search terms for an uncontrolled indexing system, and understanding of the concepts of Boolean logic are skills to be learned by the user. Representation of the search prescription as a symbolic statement and some of the terminology of information retrieval systems are also taught.

THE FILE. The file used in testing the program was a small collection of articles on the subject of "subject analysis of library materials," collected and indexed by students of the Graduate School of Library and Information Sciences. No attempt was made to control the indexing. The program operated on an inverted file of subject terms and a sequential file of article references.

DESIGN AND OPERATION OF THE GENERAL STRATEGY TUTORIAL. The program consists of two main parts which may be run independently of each other:

> The tutorial teaches the vocabulary of logical search strategy statements and illustrates use of Boolean strategies on a search question which the user has helped to create.
>
> An exercise which allows the user to create strategies for one of seven given questions, evaluates his efforts in terms of optimum results possible with the given data base, and scores him for recall and for precision.

The tutorial segment of the program guides the student user in the evolution of a vague, unformed question into a highly specific, well-developed strategy. A search for subject $A + B$ becomes successively, $A \cdot B$, $(A_1 + A_2) \cdot B$, $(A_1 + A_2) \cdot B - C$, $(A_1 + A_2) \cdot (B_1 + B_2 + B_3)$.

The exercise itself consists of two parts: (1) the query and retrieval routines and (2) the evaluation routines. Feedback to the user includes: number of hits for each search term selected; suggestions of variant forms of search terms; faulty structure of logical statement; listing of missed documents; and listing of citations and/or indexing

terms for hits, misses, or noise. Some system responses are automatic, others are optional.

Part 1 of the exercise can operate independently as a searching and reporting tool. Any body of documents and associated alphabetical indexes can be used as a data base. The size of the data base file can be expanded to the limit of disk (IBM2361) storage allotted.

Extensive diagnostic reporting in the search section of the exercise would probably make it a useful training device by itself, if it is not thought desirable to use the computer to teach the vocabulary and Boolean concepts but only the technique of organizing a logical search strategy.

EXPERIENCE IN USE OF THE GENERAL STRATEGY TUTORIAL. We have a great amount of experience with the general strategy tutorial; well over 100 individuals have used the program. The final version of the program was used in a controlled test. The subjects were graduate students enrolled in an introductory course in mechanized information handling and retrieval. Performance of the program was measured in two ways:

Pretest-posttest gain scores on an objective test were compared for 37 students who used the computer tutorial and for 50 students who attended a lecture-laboratory session on the same subject.

Assessment of strategies created by students at the end of the tutorial program.

We assumed that, if our computer tutorial taught successfully, we would find no significant difference in test scores between the computer-tutored and the lectured groups. A t-test on the mean gain scores of the two groups gave a t-score of 1.5 (critical value of t = 1.6).

Of 92 strategy statements attempted by the 37 students in this test of the program 48 were completely successful strategies—i.e., achieved precision and recall scores of 1.0. Others were varying successful—complete recall was seemingly easier to achieve than perfect precision. All were executable statements of symbolic form, using +, ·, − signs to indicate union, intersection, and complement. Such a high level of search success is a function of file size and content and of the search questions assigned.

FURTHER USE OF THE GENERAL STRATEGY TUTORIAL. The search segment of this program has been separated from the tutorial for use as an adjunct exercise for students in an introductory course in mechanized information retrieval. The development of this project, as such, is complete. No further changes in this program will be made. It will be used in conjunction with other projects in subject analysis of materials as part of the developing laboratory for information science students in the Graduate School of Library and Information Sciences. Part of the program will be adapted in the development of an interactive storage and retrieval system for the laboratory.

The original purpose of this program was to prepare the way for interactive searches of large retrospective collections. Such a program of development is planned and will derive much of its structure from the experience gained with this program.

REFERENCES

(1) Silberman, H. F., and R. T. Filep, "Information Systems Applications in Education" in *Annual Review of Information Science and Technology*, Vol. 3, p. 386, Chicago, Encyclopaedia Britannica, 1968.

(2) Morrill, Charles S., "Computer-Aided Instruction as Part of a Management Information System" in *Human Factors*, Vol. 9, No. 3, pp. 251–256 (June 1967).

(3) Cuadra, Carlos, *Technology and Libraries: A Report to the National Advisory Commission on Libraries* TM-3732, p. 129, System Development Corporation, Information Systems Technology Staff, Santa Monica, Calif. (Nov. 1967).

(4) Rubinoff, Morris, Samuel Bergman, Winifred Franks, and Elayne R. Rubinoff, "Experimental Evaluation of Information Retrieval Through a Teletypewriter," *Communications of the ACM*, Vol. 11, No. 9, pp. 598–604 (Sept. 1968).

(5) Freeman, Robert R., and Pauline Atherton, *An Experiment with an On-Line, Interactive Reference Retrieval System Using the Universal Decimal Classification System as the Index Language in the Field of Nuclear Science*, UDC Project Rept. No. AIR/UDC-7, American Institute of Physics, New York, N. Y. (April 25, 1968).

(6) System Development Corporation, "Man/Machine: A Contemporary Dialogue," *SDC Magazine*, Vol. 10, No. 9, pp. 13–19 (Sept. 1967).

(7) Burnaugh, H. P., *The BOLD User's Manual*, TM-2306/004/02, System Development Corp., Santa Monica, Calif. (June 6, 1967).

(8) Drew, D. L., R. K. Summit, R. I. Tanaka, and R. B. Whitely, "An On-Line Technical Library Reference Retrieval System," *Amer. Doc.*, Vol. 17, No. 1, pp. 3–7 (Jan. 1966).

(9) Summit, Roger K., *Remote Information Retrieval Facility*, NASA CR-1318, Lockheed Aircraft Corporation, Palo Alto, Calif. (April 1969).

(10) Cautin, Harvey, and Fredericka Rapp, *Description of Easy English*, Rept. No. 67–22, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Pa. (April 1967).

(11) Green, James S., *GRINS, An On-Line Structure for the Negotiation of Inquiries*, Studies in the Man-System Interface in Libraries, No. 4, Center for the Information Sciences, Lehigh University, Bethlehem, Pa. (Sept. 1967).

(12) Hoyt, William, "Proving it Works," *SDC Magazine*, Vol. 12, No. 2, pp. 8–13 (Feb. 1969).

(13) Meister, David, and Dennis J. Sullivan, *Evaluation of User Reactions to a Prototype On-Line Information Retrieval System*. NASA CR-918 Bunker-Ramo Corp., Canoga Park, Calif. (Oct. 1967).

(14) *Quarterly Progress Report, Jan. 1–Mar. 31, 1969, on Interactive Systems Development*, Chemical Information Experiment Station (NSF Grant **GN738**), University of Pittsburgh, Knowledge Availability Systems Center, April 1969.

(15) Caruso, D. E., *An Experiment to Determine the Effectiveness of an Interactive Tutorial Program, Implemented on the Time Sharing IBM System 360, Model 50, in Teaching a Subject-Oriented User to Formulate Inquiry Statements to a Computerized On-Line Information Retrieval System*, Ph.D. dissertation, University of Pittsburgh, Graduate School of Library and Information Sciences, 1969.

(16) *Quarterly Progress Report, July 1–Sept. 30, 1968, on Interactive Systems Development*, Chemical Information Experiment Station (NSF Grant **GN738**), University of Pittsburgh, Knowledge Availability Systems Center, Oct. 1968.

(17) *The Profile Development Program: A Tutorial for Users of the Chemical Abstracts Service Condensates Alerting Service, Reference Manual*, Chemical Information Experiment Station (NSF Grant **GN738**), University of Pittsburgh, Knowledge Availability Systems Center, Aug. 1969.