AFOSR 67-1952

AD657809

UNIVERSITY of PENNSYLVANIA

The Moore School of Electrical Engineering

PHILADELPHIA, PENNSYLVANIA 19104

D D C
RECEIVED
SEP 13 1967
C

Distribution of this document is unlimited.

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va. 22151

31

University of Pennsylvania
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING
Philadelphia, Pennsylvania

THE MOORE SCHOOL INFORMATION SYSTEMS LABORATORY

May 1967

Morris Rubinoff
Principal Investigator

S. Bergman
H. Cautin
T. Johnson
F. Franks
T. C. Lowe

J. Lucas
S. Newman
F. Rapp
E. R. Rubinoff
D. Stone

University of Pennsylvania
THE MOORE SCHOOL OF ELECTRICAL ENGINEERING
Philadelphia, Pennsylvania

THE MOORE SCHOOL INFORMATION SYSTEMS LABORATORY

The Information Systems Laboratory of The Moore School of Electrical Engineering, Unive ity of Pennsylvania, was established in 1962 to develop a design for a mec ized information system in the information processing field, with special attention to the implementation criteria entering into on-line retrieval through man-machine dialogue from a remote console. The program is currently concentrating upon four major tasks at the system level; specifications for capturing search strategies; specifications for machine storage of indexer aids, including lists of index items, synonymic equivalences, classification tables, and other semantic tools; organization of disk file storage to accommodate system routines for the load and quest modes; and study of uses of graphic display. The search mode has been implemented in minimal form and informal tests have been run. The long-range goal is to provide for machine-directed search, with computer-aided instruction on how to conduct a search, and with the search conducted in a problem-solving mode.

Out of the early reflections on the problems of indexing documents, Moore School conviction grew that information system problems stem primarily from the limitations and complexities of natural language as a means for communication. These convictions have been reinforced as the research advanced. The many attempts made by other researchers to mechanize the indexing process underscore these difficulties and suggest that they result partly from the multitude of synonymic alternatives and homographic ambiguities that pervade natural language, and partly from the omission of presumed common knowledge where the author presupposes that the reader will supply the broad framework of underlying material.

It follows that conventional schemes for cataloging and indexing are inherently limited in their ability to assist the search process. The newer procedures employ "deep indexing" techniques, whereby a substantial number of index terms, often as many as 100, are assigned to each document to supplement conventional bibliographic elements such as author, title, date, publisher, etc. As used here, an index term means a single word, number, or symbol or a brief phrase, which gives a clue to a substantial topic or item discussed in a document or denotes a subject area relevant to the document's contents.

The procedure which currently enjoys the greatest popularity makes use of an "authority list" or "thesaurus" of index terms. The thesaurus is prescribed by a group of experts in the subject specialty. These experts select a set of (relatively) independent index terms to span the topics that they believe should be included in the document file. The

primary limitation of a thesaurus is that it presupposes an arbitrary and fixed characterization of subject matter; indexing is thus restricted to these preconceived notions, and new ideas are withheld from the searcher. From the vantage point of information theory, a thesaurus restricts the apparent growth of the file to an accumulation of more and more documents on the same subject matter through extension of ideas beyond the boundaries existing at thesaurus-making time.

The Moore School research team has therefore concentrated upon the preparation of a functional and procedural plan for a mechanized information system which recognizes the limitations of index terms, whether taken freely from natural language or limited to a prescribed thesaurus. The system calls for computer aid not only in searching the document file but also in providing instructions on how the file has been organized, what index term meanings were assumed by the indexers at indexing time, which homographic meanings of index terms are allowed by the system; which synonyms are recognized, etc. In short, a "librarian" is built into the system and the user can obtain the librarian-like assistance, in real time, directly through his on-line console.

More specifically, the system is planned with the following features:

(1) the user has direct access to the system via on-line console;

(2) in addition to catalog and index data, the system will store a complete description of itself;

(3) the user will be permitted an unrestricted search vocabulary. It will be the responsibility of the system to interpret search terms, request clarification where ambiguity arises, and provide meanings of terms upon request;

(4) the user will gain access to the document file through any one or more of a large number of entry ports, such as author, data, color of document, language, etc.;

(5) the user will be able to search from an initial category through related categories, with assistance provided by the system in designating and locating related categories, terms, and other entries.

The mechanized information system has been implemented in its first form. The system has been designed with a modular structure in order that commands may be added with ease and general-purpose routines may be shared by commands. The mass store is an IBM 1301 disk storage unit accessed from an IBM 7040 computer. The jobs of editing, printing, and accepting messages from local or remote users are performed by a DEC PDP-8. Direct on-line console access is through a 33-ASR Teletypewriter; remote stations utilize the same model Teletype units with Dataphones for telephone-line connection.

Summary of Progress to Date

The procedural specifications of an information system encompass the information flow through the system from the moment that a document is retrieved through the intervals when the document is indexed, ingested, and periodically retrieved and eventually to the time that the document is purged from the system because of its obsolescence. The procedural specifications reflect the actions of individuals and groups in processing the documents and their characterization by index terms (including conventional bibliographic elements, etc.) but the responsibilities of individuals and groups are generally spelled out in the functional specifications.

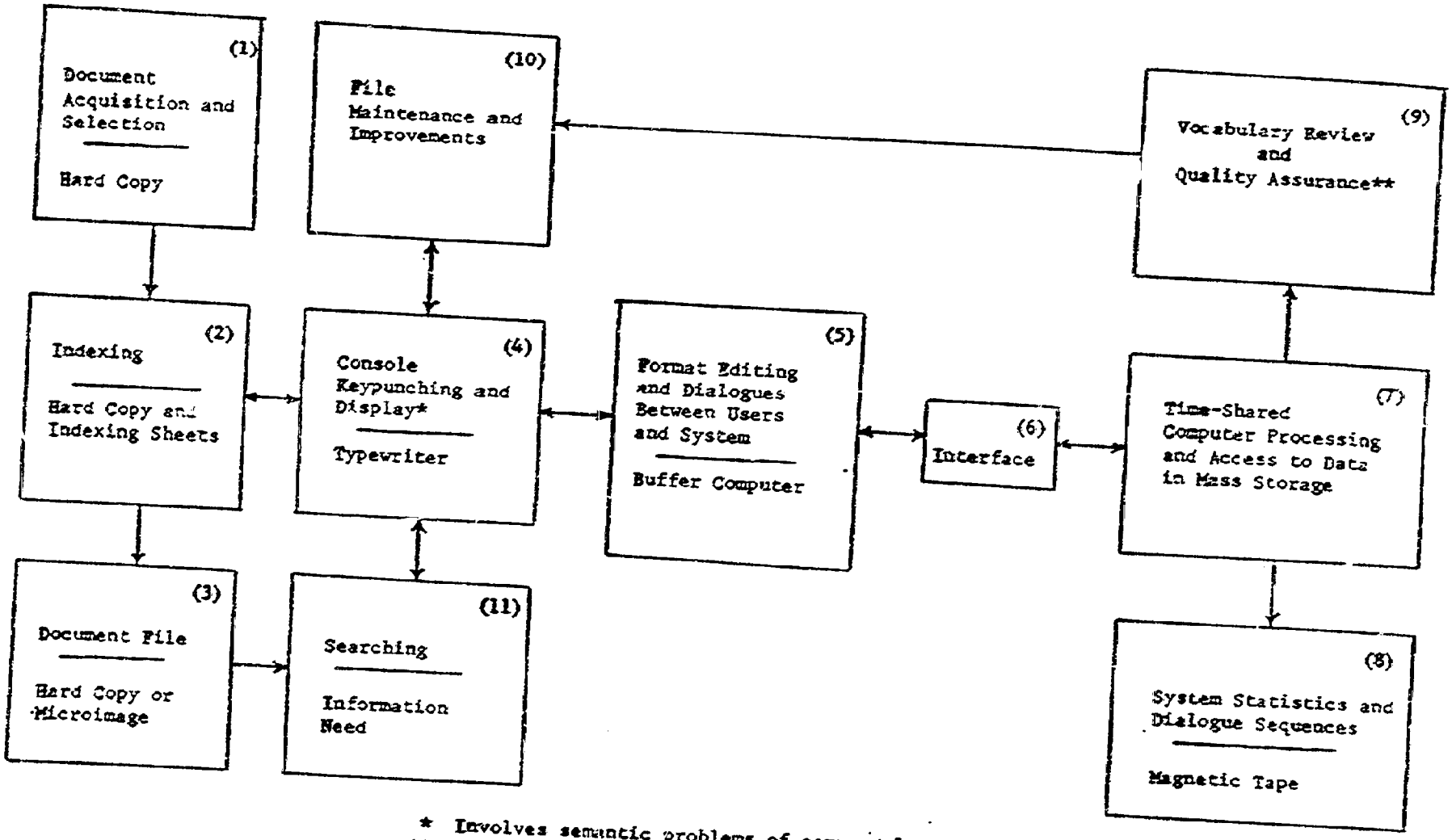
Figure 1 illustrates the information flow through the system. Each box indicates the function performed at that point and the equipment and/or form of information record. The first box depicts the arrival of new documents in hard copy form, their evaluation on the basis of prescribed criteria and their acceptance/rejection as file documents. The second box denotes the indexing of accepted documents by human indexers. The original document is then transferred to the document file (box 3) in its hard copy and/or microimage form.

The indexing sheets are forwarded to the console operator for loading into the mechanized file (box 4). Only authorized users are allowed to load new information into the files; such users are assigned special codes to identify themselves to the system and to gain access to the subroutines of the loading program.

Console keying is forwarded by Dataphone communication line (box 4) from the Moore School to the Computer Laboratory elsewhere on campus; since commercial telephone system lines are being used, any compatible teletypewriter can gain access to the system equally well. Indexing sheets and other file updating information are loaded periodically on a batch basis; this has significant influence on the file organization and the loading programs, particularly on disk file rearrangement procedures.

Messages enter the computer system through a buffer computer (box 5), where they are assembled and edited. Commands for message editing are illustrated briefly below, where one may also note the manner in which dialogue is promoted in natural English between man and machine through typewritten responses from the buffer. When the loading message and the indexing terms have been fully assembled in the buffers, a "message termination symbol" is keyed at the console which initiates transfer through a hardware interface into the computer/disk file system (box 7).

The new document index is then automatically loaded onto the disk and stored on magnetic tape. Deviations from the standards, such as index terms that do not appear in the descriptor lexicon, are summarized and forwarded to the vocabulary review board (box 9) where instructions are generated for file maintenance and improvements (box 10). The latter include expansion of the lexicon of transients, promotion of transients to free terms and free terms to descriptors, adjustment of spelling errors,



* Involves semantic problems of command language
** Involves semantic problems of data/descriptors

Information Flow in a Remote-Access Information System

Figure 1

etc. The review board instructions are then implemented through the console, also in the privileged load mode.

Certain system statistics are summarized periodically. These include frequency of descriptor usage, extremely high or low document activity, difficulties in man-machine dialogue, etc. The summarized statistics are reviewed by the quality assurance board which writes instructions for system modification including descriptor demotion, document purging, and adjustments to console commands and the retrieval language. These instructions are also entered via the console in the load mode.

The user gains access to the system when the teletypewriter console is free for user search (box 11). The user is only permitted to use the search mode to access the system. Suggested system improvements can be inserted into his search sequence but these are simply stored on magnetic tape (box 8) for later review by the quality assurance board.

About 2000 documents have been manually indexed. All have been keypunched onto Hollerith cards and are loaded onto disk for search experiments. The programs have been written for deriving inverted lists of accession numbers, which facilitate rapid retrieval through logical combinations of index terms. An executive control system has been written which interprets search commands, corrects minor spelling errors in the search message, and provides a number of similar user services.

Operation of the system may be better understood, at least in part, by reference to the results of an actual search displayed in Figure 2. The print-out begins with a succession of lines of STANDBY; these were typed by the buffer computer at one-minute intervals to assure the searcher that his console was still connected but that the IBM 7040 had not yet been turned over to the system by the Computer Center personnel. Possession is finally indicated by the buffer notification, I AM :- .

The searcher gave his code number (indicating character erasure with a left arrow) and the buffer requested mode designation. SEARCH was recognized as an existing mode, and the searcher was permitted to proceed.

The searcher requested a combinational search (keyword: COMBINE) on four terms, three of which were descriptors ($B) and one a date ($A2). After message printout, a correction was made by the searcher who added another term to the request. Finally after another message printout, retrieval was implemented and the searcher informed of the results. The buffer automatically asked if catalog data on these documents were desired, with "DO YOU WANT TO SEE ANY?"

Affirmative response was followed by a sequence of queries to determine which of the documents and which catalog data were desired. Note that the machine directed the searcher at this point. In fact, the searcher was interested in accession numbers only, and all 37 he requested were printed out.

In the next man-machine dialogue (Figure 2c), the RETRIEVE command

Figure 2a--An Illustrative On-Line Search

```
STANDBY.
STANDBY.
STANDBY.
STANDBY.
STANDBY.
STANDBY.
STANDBY.
STANDBY.
STANDBY.
STANDBY.
STANDBY.
STANDBY.

I AM :=                    003W-333<>

THE OPERATING MODE IS :=                    SEARCH<>

YOU MAY PROCEED.:=
    MANUAL/FEB ----$A8 FEB 1955<>COMBINE $B IBM/PROGRAMMING/

PRINT? :=    YES<>

YOUR MESSAGE IS:

01) COMBINE $B IBM/PROGRAMMING/
02) MANUAL/$A8 FEB 1955


CORRECTIONS? :=    YES<>

LINE NO.  :=    3<>
LINE(S):

    /FORT-?
LINE(S):

    /$B FORTRAN<>

MORE? :=    NO<>
PRINT? :=    YES<>
```

Figure 2b - An Illustrative On-Line Search

```
YOUR MESSAGE IS:

01) COMBINE $0 IBM/PROGRAMMING/
02) MANUAL/$A2 FEB 1955
03) /$B FORTRAN


CORRECTIONS?  := NO<>

DOCUMENTS HAVE BEEN RETRIEVED AS FOLLOWS:
000000 DOCUMENTS CHARACTERIZED BY 5 ITEMS.
000000 DOCUMENTS CHARACTERIZED BY 4 ITEMS.
000037 DOCUMENTS CHARACTERIZED BY 3 ITEMS.
000190 DOCUMENTS CHARACTERIZED BY 2 ITEMS.
000328 DOCUMENTS CHARACTERIZED BY 1 ITEM.

DO YOU WANT TO SEE ANY? := YES<>

DOCUMENTS CHARACTERIZED BY AT LEAST HOW MANY ITEMS? := 3<>

INDICATE SECTOR INFO. DESIRED. (ANSWER 'YES', 'NO', 'ALL', OR 'FORGET').
ALL $A? := NO<>

ANY $A? := NO<>

$B? := <>

PLEASE ANSWER 'YES','NO','FORGET' OR 'ALL') := NO<>

$C? := NO<>

ACCESSION NUMBERS FOUND:

100  101  102  103  104  105  106  107  108  113  114
125  126  127  134  141  157  163  168  169  170  171
172  173  176  180  182  187  197  200  204  222  231
236  270  70   71

                    THAT'S ALL.
```

Figure 2c - An Illustrative On-Line Search

```
YOU MAY PROCEED.:=                    RETRIEVE $A: PATTERSON
    + CARR, J. W. <>

PRINT?  :=     NO<>

DO YOU MEAN      RETRIEVE    ? :=      YES<>

000004 'REFERENCES' HAVE BEEN RETRIEVED.

PRINT SOME?  :=     YES<>

SAME INFORMATION CATEGORIES AS BEFORE?  :=    NO<>

INDICATE SECTOR INFO. DESIRED. (ANSWER 'YES', 'NO', 'ALL', OR 'FORGET').

ALL $A?  :=     YES<>

 $B?  :=    NO<>

 $C?  :=    NO<>

 ACC. NO.:  1
A0 8/3/65+JB
A2 JUNE 1954
A3 FIRST GLOSSARY OF PROGRAMMING TERMINOLOGY- REPORT TO THE ASSOCIATION
A3  FOR COMPUTING MACHINERY
A1 ADAMS,C W+BACKUS,J W+CARR,J W III+OSBORN,R F+PATTERSON,G W+SVIGALS,J+
A1 WEGSTEIN,J+HOPPER,GRACE MURRAY
A5 ASSOCIATION FOR COMPUTING MACHINERY, NEW YORK
A6 17X25 CM
A7 0
A8 PP 25+8

 ACC. NO.:  40
A0 8/5/65+JB
A1 PERKINS,ROBERT+CARR,JOHN W III+BROWN,J HARVEY
A2 14 SEPT 1955
A3 EASIAC, A PSEUDO COMPUTER-- A PAPER PRESENTED AT THE ANNUAL MEETING O
A3 F THE ACM, PHILADELPHIA, PA, 14 SEPT 1955
A5 RAMO-WOOLDRIDGE CORP, 8820 BELLANCA AVE, LOS ANGELES 45, CALIF
A6 22X28 CM
A7 4
A8 PP 9+1

MORE?  :=     NO<>
```

Figure 2d - An Illustrative On-Line Search

```
YOU MAY PROCEED.:=                    DISPLAY (100,127,187,270)
    <>

PRINT?  :=    NO<>

SAME INFORMATION CATEGORIES AS BEFORE?  :=    NO<>

INDICATE SECTOR INFO. DESIRED. (ANSWER 'YES', 'NO', 'ALL', OR 'FORGET').

ALL $A?  :=    NO<>

ANY $A?  :=    YES<>

GIVE SECTOR DIGITS  :=    1,3<>

 $B?  :=    NO<>

 $C?  :=    NO<>

 ACC. NO.:  100
A3 IBM REFERENCE MANUAL- 704 FORTRAN PROGRAMMING SYSTEM

 ACC. NO.:  127
A3 IBM 704 AND 709 DATA PROCESSING SYSTEMS BULLETIN- 704 AND 709 FORTRAN
A3 - USING FUNCTION AND SUBROUTINE NAMES AS ARGUMENTS

 ACC. NO.:  187
A3 IBM 7030 DATA PROCESSING SYSTEM- IBM 7030 SYSTEMS PROGRAM PACKAGE

 ACC. NO.:  270
A1 KATZ,CHARLES
A3 COMPARATIVE CODING FOR FORTRAN (IBM 704), MATH-MATIC (UNIVAC I AND II
A3 ), UNICODE (1103A)

                                THAT'S ALL.

YOU MAY PROCEED.:=                    END<>

PRINT?  :=    NO<>

YOU HAVE GIVEN THE END SIGNAL.

CONNECTION TERMINATED.
```

was employed, allowing for any logical combination of request terms. The union of documents by two authors was requested, and again, the searcher was directed to specify catalog data desired. Note that the searcher misspelled RETRIEVE and was corrected. As soon as format was determined the machine printed out two documents. More could have been obtained by answering YES to the machine query MORE?.

In the last dialogue (figure 2d), the DISPLAY command was used in order to have certain catalog data of specific documents printed out. The rest of the dialogue should be self-explanatory. The message END terminated the connection.

Easy English

The commands RETRIEVE and COMBINE illustrate the operation and behavior of Symbolic Command Language (SCL) as a means for man/machine communication through a typewriter console. SCL proved satisfactory for users of the information system who participated in the study or who happened to wander in during demonstrations. It was therefore decided to perform an experiment with a more universal set of subjects who were less skilled with mathematics and computer languages than the study participants and casual visitors. The new subjects consisted of secretaries and undergraduate students, and their poor results deflated the confidence in SCL that had been building up.

A far-reaching action was then taken, to leapfrog ahead, rejecting all artificial languages and turning instead to a somewhat restricted but nevertheless real version of English. Fortunately, as mentioned above, SCL was designed in modular form and it has been found possible to set the new command language, "Easy English", directly over the top of SCL which it then uses for the actual search. The following is a summary description of the new command language, which has been fully operational since February 1967.

Easy English is a plain command language designed to simplify dialogues between man and machine through a remote typewriter console. It is made up of readily recognized sentences of the English language, sentences which any layman might be expected to use in everyday requests for services or articles from a familiar source. Easy English has been developed as a command language for retrieval of documents from a computerized data base, specifically from the Moore School Information Systems Laboratory (MSISL) files. It is intended for all information retrieval systems using remote typewriter consoles in a conversational mode.

Easy English is imbedded in the MSISL retrieval program which provides computer-directed search, computer-aided editing, and other forms of computer assistance. The attached typewriter printout presents a typical man-machine conversation which illustrates Easy English along with a number of features of the Laboratory retrieval system. Note that the latter currently provides the option of translation of the Easy English request into Symbolic Command

Language while searching the files; this is a convenience for those who might like to learn Symbolic Command Language on their own and use its shorter but more formal statements in place of Easy English.

Because Easy English is in fact real English, the only thing that the searcher needs to learn is that requests for information from the system should be formulated in the following syntactical form:

Introductory Clause Document Clause Data Clause .

The following sentences present five forms in which the same retrieval request can be phrased in Easy English.

(1) PLEASE LOCATE EVERYTHING WRITTEN BY ROBERT PERKINS ABOUT EASIAC OR PSEUDO-COMPUTERS BETWEEN 1955 AND 1959 < >

(2) COULD YOU FIND FOR ME SOMETHING CONTAINED IN THE REPOSITORY CONCERNING EASIAC OR PSEUDO-COMPUTERS THAT WAS AUTHORIZED BY ROBERT PERKINS AFTER 1954 AND BEFORE 1960 < >

(3) I NEED ALL THE AVAILABLE DOCUMENTS PUBLISHED DURING THE PERIOD 1955 TO 1959 BY ROBERT PERKINS ON THE SUBJECTS OF EASIAC OR PSEUDO-COMPUTERS < >

(4) WE'RE INTERESTED IN HAVING REFERENCES AND MATERIAL ON EITHER PSEUDO-COMPUTERS OR EASIAC AUTHORED BY ROBERT PERKINS FROM 1955 TO 1959 < >

(5) I WOULD LIKE YOU TO HELP ME OBTAIN INFORMATION FROM YOUR LIBRARY RELATED TO EASIAC OR PSEUDO-COMPUTERS AND WRITTEN BY ROBERT PERKINS IN THE YEARS 1955 THROUGH 1959 < >

Notice that despite the differences in vocabulary, all of these statements follow the same basic pattern; for example,

COULD YOU FIND FOR ME SOMETHING CONTAINED IN

THE REPOSITORY CONCERNING ...

Typical examples of phrases acceptable in the three clause categories are:

<u>Introductory clause</u>

(1) I would like ...
(2) Please find for me ...
(3) I have need of ...
(4) I desire ...

Document clause

(1) ... documents in the system ...
(2) ... information ...
(3) ... any available book or article in the repository ...
(4) ... references from the files ...
(5) ... all the stuff ...

Data clause

(1) ... written by Carr between 1958 and 1965.
(2) ... published in 1960 on information retrieval and word association but not programming.
(3) ... dated September 1966 by J.H. Smith, Joe Doe but not K.L. Jones about analog computers.

In the event that a word appearing in either the introductory or the document clause is not recognized, the computer initiates a man-machine dialogue in order to determine whether the word is essential and, if so, to seek out a synonym in its vocabulary. Examples of such dialogues appear on the attached typewriter printout.

Current Tasks

With the system operational in its first form, attention has been directed to accommodation of the many other features implied by the procedural flow chart in Figure 1. A number of tasks have been defined and are described below. Documentation of the mechanization to date has been initiated on three levels: detailed microflowcharts, programs, and related descriptive text; macroflowcharts and related text describing the gross features of the system and showing the interrelationships among the detailed programs; and brief summaries of the main features of the system, cross-referenced to the macroflowcharts.

As mentioned above, the Moore School information system plan calls for computer-aided search. The basic tactic for computer aid is a sophisticated use of man-machine dialogue in a problem-solving mode. The system incorporates computer-aided instruction as a means for assisting the searcher to find not only information about the document file but also information on how to go about the search, i.e., computer-aided instruction on computer-aided search. At the same time, the Moore School recognizes the services provided by conventional bibliographic tools, and particularly those which expedite search through condensed display of bibliographic coupling. The tasks reflect this broad range of information system support.

The tasks may be delineated in the following order, proceeding from the simplest aids to the most advanced:

- Syntactic Tools

```
YOU MAY PROCEED:=    PLEASE FIND FOR ME BOOKS
    CONCERNING STATISTICAL FUNCTIONS OR STANDARD
    DEVIATION, BUT NOT BUSINESS ORIENTED ZILKS
    INC***ENTITLED 'RUNCIBLE''I'.<>
```

{ Message is entered.

```
PRINT?  :=    YES<>

YOUR MESSAGE IS:

01) PLEASE FIND FOR ME BOOKS
02) CONCERNING STATISTICAL FUNCTIONS OR STANDARD
03) DEVIATION, BUT NOT BUSINESS ORIENTED ZILKS
04) ENTITLED 'RUNCIBLE''I'.
```

{ Edited message is printed out with lines enumerated by computer.

```
CORRECTIONS?  :=    YES<>

LINE NO.  :=    3<>
LINE(S):

    DEVIATION, BUT NOT BUSINESS ORIENTED<>
```

{ A line is changed.

```
MORE? :=    YES<>

LINE NO.  :=    0<>
LINE(S):

    DEAR COMPUTER:<>
```

{ A line is added.

```
MORE? :=    NO<>
PRINT?  :=    YES<>

YOUR MESSAGE IS:

01) DEAR COMPUTER:
02) PLEASE FIND FOR ME BOOKS
03) CONCERNING STATISTICAL FUNCTIONS OR STANDARD
04) DEVIATION, BUT NOT BUSINESS ORIENTED
05) ENTITLED 'RUNCIBLE''I'.
```

{ This is the final request in Easy English.

```
CORRECTIONS?  :=    NO<>

WE DID NOT RECOGNIZE THE WORD DEAR    .IS THIS WORD ESSENTIAL TO
    THE MEANING OF YOUR SENTENCE? :=    NO<>
```

A non-essential word is challenged and then ignored.

```
WE DID NOT RECOGNIZE THE WORD COMPUTER     .IS THIS WORD ESSENTIAL TO
  THE MEANING OF YOUR SENTENCE? :=     NO<>

RETRIEVE          (  $B        (      (STATISTICAL FUNCTIONS      {Symbolic
     )      +     (STANDARD    DEVIATION          )              Command
     )      !     (BUSINESS    ORIENTED           )              Language
     )      &     ( $A3  RUNCIBLE            )                   Equivalent
     )

000001 'REFERENCES' HAVE BEEN RETRIEVED.

YOU MAY PROCEED.:=      PLEASE GET BOOKS BY--WRITTEN, EDITED {Another
    OR PUBLISHED BY CARR.<>                                   {request in
                                                              {Easy English.

PRINT?  :=    NO<>

RETRIEVE          (      ( $A1  CARR        +  $A4  CARR        )
     +      ( $A5  CARR        )      )

000009 'REFERENCES' HAVE BEEN RETRIEVED.

PRINT SOME?  :=     NO<>

YOU MAY PROCEED.:=      GET BOOKS BY EITHER CARR OR RUBINOFF {A Third
    BUT NOT BY CARR<>                                        {request.

PRINT?  :=    NO<>

RETRIEVE          ( $A1        (       CARR        )      +      (
RUBINOFF          )      )      !      ( $A1  CARR
     )

000001 'REFERENCES' HAVE BEEN RETRIEVED.

YOU MAY PROCEED.:=      OBTAIN FOR ME BOOKS WRITTEN IN 1961 <>
                                                             {Request
                                                             {number 4.

PRINT?  :=    NO<>

RETRIEVE      $A2  1961

000127 'REFERENCES' HAVE BEEN RETRIEVED.

PRINT SOME?  :=     NO<>

YOU MAY PROCEED.:=      I WOULD LIKE YOU TO FIND BOOKS   {Request number 5.
    WRITTEN, EDITED, AND PUBLISHED BY CARR.<>            {Note the ability
                                                         {of the system to
                                                         {separate out the
                                                         {author, editor, and
PRINT?  :=    NO<>                                       {publisher functions.

RETRIEVE          (      ( $A1  CARR        &  $A4  CARR        )
     &      ( $A5  CARR        )      )

NO 'REFERENCES' HAVE BEEN RETRIEVED.
```

- Semantic Tools
- Indexer Aids
- Graphic Display
- Adaptive Interface
- Intersystem Switching

A few comments will be made on each item.

Syntactic tools are those which make use of word associations or author-designated couplings. Specific examples are permuted title indexes (KWIC) and citation indexes. The program to derive a KWIC index of the collection has been prepared, and citation index preparation has been initiated. Both will be printed for visual use in the Moore School library; KWIC is automatically available in the mechanized system and the citation index will be added.

KWIC does not pretend to be a sophisticated cataloging system, but its use in association with a document library has the following advantages:

1. It is an inexpensive way to produce a printed catalog of the library, since it can be automatically produced by the system in a format ready for photo reduction and offset printing.

2. It is easy to keep up to date, as the documents are already indexed and stored in the mechanized system.

3. It can be widely distributed in simple loose-leaf or book form.

4. It provides a printed record of the documents stored in the system at any time in its development.

It is conjectured that a citation network, connecting every document in a library to every other one which cites it or is cited by it, might be a useful tool. Various studies will be made of experimental citation networks, with the goal of adding this tool to the system. The citation network might be made directly available to the user, so that he may start at any document and follow a trail of citations either forward or backward in his document search. More important, manipulations of a citation network may be applicable to adaptive interface techniques for unsolicited machine assistance to the searcher.

Research has been initiated to investigate the uses and effectiveness of word associations to a greater depth than previously attempted by other experimenters. An important shortcoming in earlier attempts was their failure to distinguish among various types of intellectual coupling that contributed in equal amount to the derived association factor. For example,

if index terms (A, E) are strongly associated, and (B, E) are strongly associated, but (A, B) are weakly associated, there are at least two interpretations which have opposite implications for information retrieval. The first interpretation is that A and B are synonyms and authors have subjective preferences for one or the other. The second interpretation is that E is a homograph with two different meanings, A and B, and that the latter have no intellectual ties whatsoever. The first task will be therefore to seek out the various intellectual relationships that may be associated with different patterns of word associations.

It is important to note that word associations could serve information systems in a number of ways. In establishing a system, word associations readily lend themselves to vocabulary synthesis, to document indexing, and to determination of relevance of a document to the scope of the file. In system search, word association lends itself to extension of search scope by searching on unsolicited but strongly associated words.

Semantic tools include a number of devices, such as sets of synonymic equivalents, classification tables based on a variety of word relationships, and semantic expansions. These tools are applicable both to index terms and to system commands. Synonyms are used even on conventional thesauri, but it is intended to expand their use and provide automatic substitution where use of a preferred synonym leads to system efficiencies. Classification tables (Figures 3a-3f) provide tools for browsing through the indexing structure, discovering relationships among words, branching between unrelated topics through homographic coupling, etc. And semantic expansions provide explanations and illustrations at several levels of detail which serve to instruct the searcher on the meanings of words and their mode of employment in the system. For example, consider the following semantic expansion of the descriptor INTERPRETIVE PROGRAM, supposing that a searcher has asked that this term be explained for him and that he requests further explanation after each of the first three system responses. (Underlining indicates that explanation of the underlined term is also available from the system.)

A. First-level response:

"An INTERPRETIVE PROGRAM is a computer program that combines translation and execution."

B. Second-level response:

"An INTERPRETIVE PROGRAM is a computer program which receives a sequence of commands in a source language, examines each command, determines a translation to replace it in the object language, and executes it if possible. The major characteristic of an INTERPRETIVE PROGRAM is that the translation of an instruction is performed each time the instruction is to be obeyed."

C. Third-level response:

"An INTERPRETIVE PROGRAM carries out the instructions of a program written in one language by translating each instruction

TERM \ SYNONYMS	Assembler	Assembly Routine	Command	Computer Code	Computer-Dependent Language	Function	Generator	Instruction Code	Interpreter	Interpretive Routine	Machine Instruction	Machine Language	Machine Operation	Machine-Oriented Language	Operation Part	Order	Order Code	Procedure	Pseudo-Instruction	Routine
Assembly Program	X	X																		
Computer Instruction			X								X					X				
Computer Instruction Code				X				X				X					X			
Computer Language					X							X		X						
Computer Operation						X							X							
Computer-Oriented Language					X							X		X						
Directive			X								X					X			X	
Generating Program							X													
Instruction			X								X					X				
Instruction Set								X									X			
Interpretive Program									X	X										
Operation Code				X											X		X			
Program																		X		X

Condensed Sample of Classification Table for Synonyms
(from table approximately six times this size)

Figure 3a

TERM \ WORDS IN DEFINITION	Code	Computer	Construction	Elementary	Instruction	Language	Operation	Program	Programming	Specification	Symbol	Translation
Assembly Program	X					X	X	X			X	X
Computer Language		X			X	X	X		X			
Generating Program		X	X					X			X	
Instruction		X		X						X	X	
Operation Code	X	X		X			X			X	X	

TERM-WORDS IN DEFINITION

Figure 3b

GENERIC \ SPECIFIC	Assembly Language	Assembly Program	Auto Code	Computer Instruction	Directive	Instruction Code	Interpretive Program	Jump Command	Operation Code	Problem-Oriented Language	Procedure-Oriented Language	Programming Language	Pseduo Code	Translating Program
Code			X			X			X				X	
Command								X						
Instruction				X	X									
Language										X	X	X		
Program		X					X							X
Programming Language	X		X										X	

GENERIC-SPECIFIC

Condensed Samples of Classification Tables

Figure 3c

WHOLE \ PART	Assembly Program	Computer Instruction	Diagnostic Program	Generating Program	Instruction	Operation	Picture	Pragmatics	Semantics	Statement	Symbol	Syntax
Compiling Program	X		X	X								
Computer Instruction						X						
Computer Language		X										
Generating Program	X											
Language					X			X	X	X	X	X
Programming Language					X		X			X		

WHOLE-PART

Figure 3d

NOUN \ MODIFIERS	Absolute	Assembly	Automatic	Complete	Computer	Diagnostic	Instruction	Interpretive	Machine	Macro	Operation	Order	Problem-oriented	Programming	Translating
Code					X		X		X		X	X			
Instruction	X			X	X				X	X					
Language		X			X				X				X	X	
Operation			X		X				X						
Program		X			X	X		X		X					X

NOUN-MODIFIERS

Condensed Samples of Classification Tables

Figure 3e

MEANS \ END	Construction	Control	Description	Execution	Interpretation	Preparation	Production	Representation	Specification	Translating	Unconditional transfer
Assembly Program					X		X			X	
Computer Instruction				X					X		
Directive		X									
Interpretive Program				X						X	
Jump Command											X
Operation Code								X	X		

MEANS-END

Condensed Sample of Classification Table

Figure 3f

of that source language into a sequence of computer instructions in the language of the computer being used, and by allowing these computer instructions to be obeyed before translating the next instruction. This step-by-step translation and execution becomes significant when the execution of one instruction causes a change in the meaning of that instruction or another one. A new translation of the changed instruction will be necessary then before it can be correctly executed."

D. Fourth-level response:

"Consider the following sequence of instructions:

Position	Instruction
1	FETCH 5
2	ADD 6
3	STORE 5
4	GO TO 1
5	...
6	...

An INTERPRETIVE PROGRAM might first translate FETCH 5 into 24005 ("bring into the accumulator the contents of memory position 5") and execute the instruction. Then it might translate ADD 6 into 28006 ("add to the accumulator the contents of memory position 6") and execute that instruction. Then it might translate STORE 5 into 02005 ("store the contents of the accumulator in memory position 5") and execute that instruction. Finally, it might translate GO TO 1 into 32001 ("go to the instruction located in memory position 1 and execute it"). The instruction located at memory position 1 is FETCH 5. Because the INTERPRETIVE PROGRAM has carried out all instructions immediately after translating them, memory position 5 now contains a new value which will be incorporated into all further instructions involving it. If translation of all instructions had been completed before any of them had been executed, such a change would have been ignored. This demonstrates the major characteristic of an INTERPRETIVE PROGRAM -- that translation of an instruction is performed each time the instruction is to be obeyed."

While such semantic tools as classification tables and semantic expansions were originally conceived and developed by the Moore School as searcher aids, it appears that they are also necessary in the areas of vocabulary control and document indexing. As mentioned earlier, much of

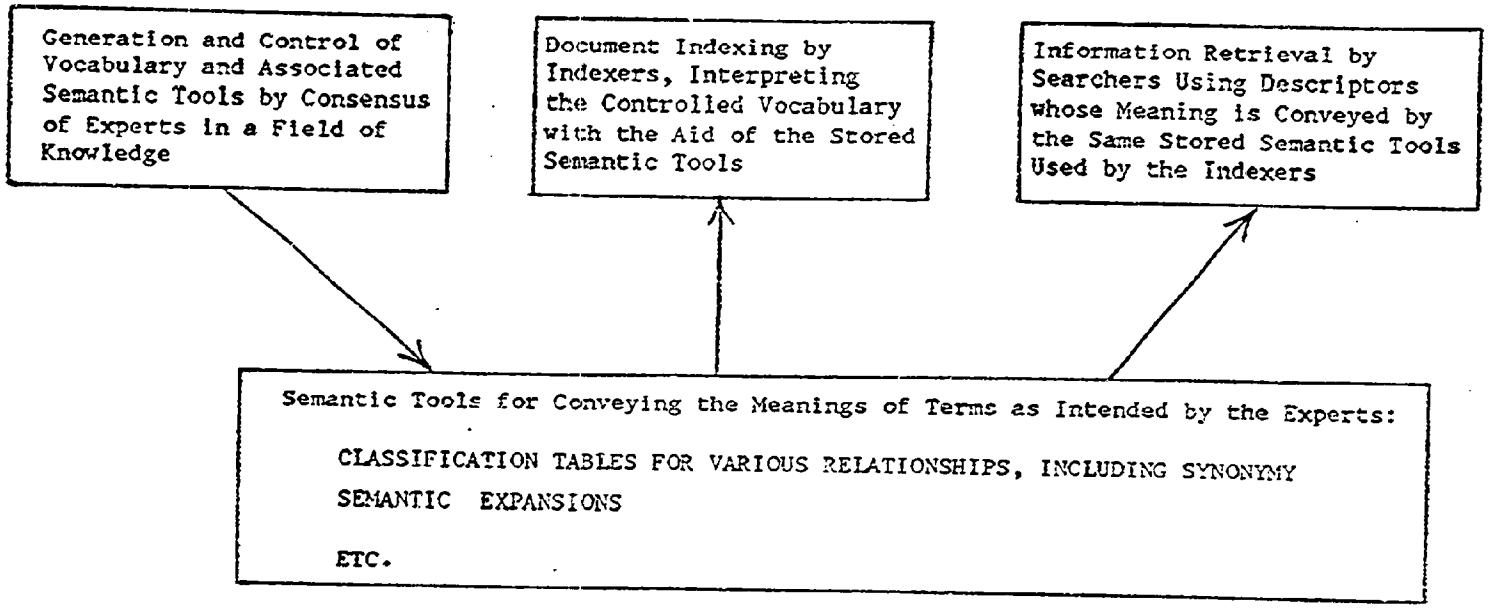
the search process is dependent upon the quality of indexing, and indexing quality is likewise dependent upon the quality of the thesaurus from which items are chosen. The Laboratory has concluded that research and implementation of semantic tools are necessary to the development of better mechanized information systems. Figure 4 illustrates the role that semantic tools play in the interactions that exist among those who devise and control the vocabulary and those who use it for indexing and searching.

It should be recalled that the system aims to provide a broad cross-section of users with access to the file of documents. This requires a system which is self-explanatory to a user who has no experience with a mechanized library, and is only superficially familiar with usual library techniques, including the availability of a librarian's guidance. The capability of "carrying on a conversation" with the user will explain, instruct, volunteer suggestions, and guide. These semantic tools, and especially semantic expansion, will incorporate ideas from the field of programmed instruction (teaching machines), specifically techniques of what is known as "intrinsic programming". At a superficial level this means that any user may request instruction in the use of any or all phases of the system. If simple information does not satisfy him, he may request more detailed information, first about the general organization of the system, and then about the actual structure and inner workings of any or all of its substructures. "Semantic expansion" techniques will afford the user any degree of detail he may desire (see Figures 5a and 5b for a semantic expansion of the system command RETRIEVE). This feature of the mechanized system corresponds to a librarian's ability to explain how a library is organized or structured, either superficially, or in great detail.

At present, all <u>document indexing</u> is done manually. This introduces the difficulty of searching for indexing terms that may have been used previously and would serve again. A second difficulty stems from the delays and errors introduced in the multi-step process of writing down the index terms, sending the indexing sheets to be keypunched, listing the results on a high-speed printer, and then comparing against the original. Both difficulties can be alleviated by providing the indexer with an on-line direct-access console.

Initially, the mechanized system will be provided with the capability of accepting new document information directly from the indexer. It will request the desired information item by item, and it will accept this information only in a standard format, rejecting ambiguous entries. Eventually, by means of the semantic tools mentioned earlier, the system will assist the indexer with his choice of subject index and secondary index terms, assuring greater standardization of index terminology and omission of extraneous material.

<u>Graphic</u> display consoles offer many advantages over teletypewriters as terminal units. Currently in use at the Moore School are two Bunker-Ramo Teleregister consoles which allow character display only. More sophisticated devices, permitting diagram display and light pen operation are currently being developed. Such devices have great potential in the information laboratory, as they free the system design from many constraints imposed by



Communication Between Experts, Indexers, and Searchers Through Semantic Tools, in an Information Retrieval System with a Controlled Vocabulary

Figure 4

Semantic Expansion of the Command RETRIEVE

[N.B. Underlining indicates that further explanation is available from the system by specifying the underlined word or word phrase.]

A. First-level response:

"RETRIEVE is a command used to obtain information about documents in the data base, according to the specifications of a user of the system."

B. Second-level response:

"The RETRIEVE command provides information about documents in response to given index term specifications. You may specify the required document(s) by bibliographic data (category $A with sector codes $A1 - $A9) , descriptors (category $B), or added information (category $C). Your request will be formulated by the following logical combinations of your specifications:

a) documents characterized by all of the terms (and, &)

b) documents characterized by at least one term (or, +)

c) documents characterized by all of some terms but not by one or more other terms (and not, ↑).

When you have given a RETRIEVE command, you will be informed how many documents have been retrieved and will be asked to specify the types of information (accession numbers, $A, $B, $C) you would like to have. The requested information for each retrieved documents will then be printed out."

C. Third-level response:

"Consider the RETRIEVE command

RETRIEVE ($A1 CARR, J W III + $B PROGRAMMING LANGUAGES) ↑ IBM MANUALS < >

The machine will respond to this instruction by finding the accession numbers of all documents having J. W. Carr, III, as an author ($A1) and "Programming Languages" but not "IBM Manuals" as descriptors ($B). The machine will type out the number of documents satisfying these conditions, offer to PRINT SOME?:= , and, if your answer is YES, ask which categories of the documents' full description (accession numbers, $A, $B, $C) you would like to see. The machine will then type out the appropriate information stopping periodically to ask MORE?:=. When you answer NO < > at any time during this sequence or when all the requested information has been printed out, the machine will indicate readiness to receive a new instruction with YOU MAY PROCEED:=.

D. Fourth-level response:

See Flowchart and Table for RETRIEVE Command.

Figure 5a

Figure 5b

Results following from actions (on left) under the indicated conditions:

Actions by User or System	Always Follows	User answers or system finds: Yes	No	All*	Forget*
1. User types RETRIEVE command such as RETRIEVE $A1 RUBINOFF & $B AUTOMATA < > using logical combinations (&, +, ↑) of the specifications (author, title, index terms, etc.) of the documents he needs with the specifications prefixed by their appropriate category symbols ($A1-$A9, $B, $C).	2				
2. System finds documents answering specifications of user.		5	3		
3. System informs user that NO DOCUMENTS HAVE BEEN RETRIEVED	4				
4. System types YOU MAY PROCEED.					
5. System informs user how many documents have been retrieved; asks PRINT SOME?	6				
6. User answers question in (5).		7	4		
7. System asks user to specify desired information categories by answering questions about each and asks user to answer YES, NO, ALL or FORGET to each question.	8				
8. User answers questions in (7).		9	13**	9	4
9. System prints out the desired information for all documents found.	10				
10. User answers question in (9).		11	4		
11. System prints all desired information for all documents found.	12				
12. System types THAT'S ALL.	4				
13. System prints out accession numbers for all documents found, stopping periodically to ask MORE?	14				
14. User answers question in (13).		15	4		
15. System prints accession numbers for all documents found.	12				

* Condition applies only to action 8 ** Result follows only if "NO" is only condition

Table (Fourth-level Response in Semantic Expansion of RETRIEVE Command)

the ten-character-per-second teletype. On many occasions in man-machine dialogue, the searcher is offered a choice among a number of alternatives, each alternative leading to another sequence of choices. A video console allows simultaneous display of all alternatives and the wherewithall for a better decision. Furthermore, text and catalog data can be instantly displayed. The light-pen feature invites rapid reply by the searcher. Similarly, on-line indexing is facilitated by display of portions of microthesauri and classification tables. Flowcharts like that in Figure 6 replace the harder-to-read tables (Figure 5b) in semantic expansions. These and other advantages of graphic display will be considered in the current design and implementation.

"Adaptive interface" refers to the system capability of unsolicited suggestion and assistance to the user. This capability will probably be comprised of a large variety of techniques and will be employed in many modes of operation of the system. It may be compared to the unsolicited assistance offered by a librarian who has come to understand the problem of a library user through his questions. The librarian might say: "You are asking the wrong kind of questions, so let me suggest ...". Much of the adaptive interface technique will probably be based on results of the study of search strategies of various users. Pragmatic experiments will be set up. Choice of subjects will take account of their background, motivation, and adaptability with respect to the mechanized system. Use will be made of the large amount of literature in the field of "artificial intelligence", including work in machine problem-solving and self-organizing systems. The work being done in word association and citation networks will also come into play.

A flowchart delineating the organization and sequence of tasks is shown in Figure 7. Implementation tasks are supported by Army Research Office (Durham); system studies are supported by the Air Force Office of Scientific Research.

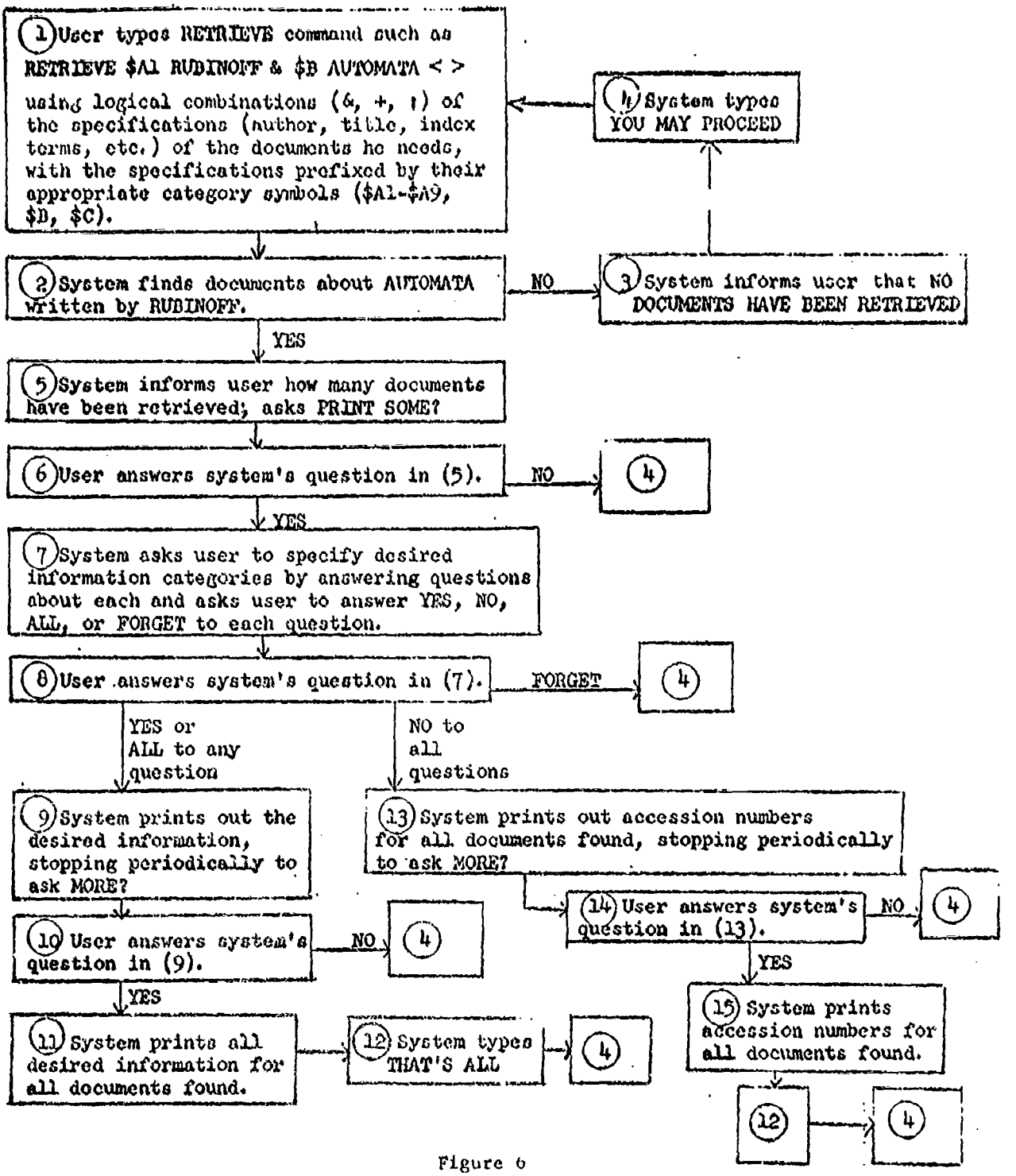


Figure 6

FLOWCHART

(Fourth-level Response in Semantic Expansion of RETRIEVE Command)

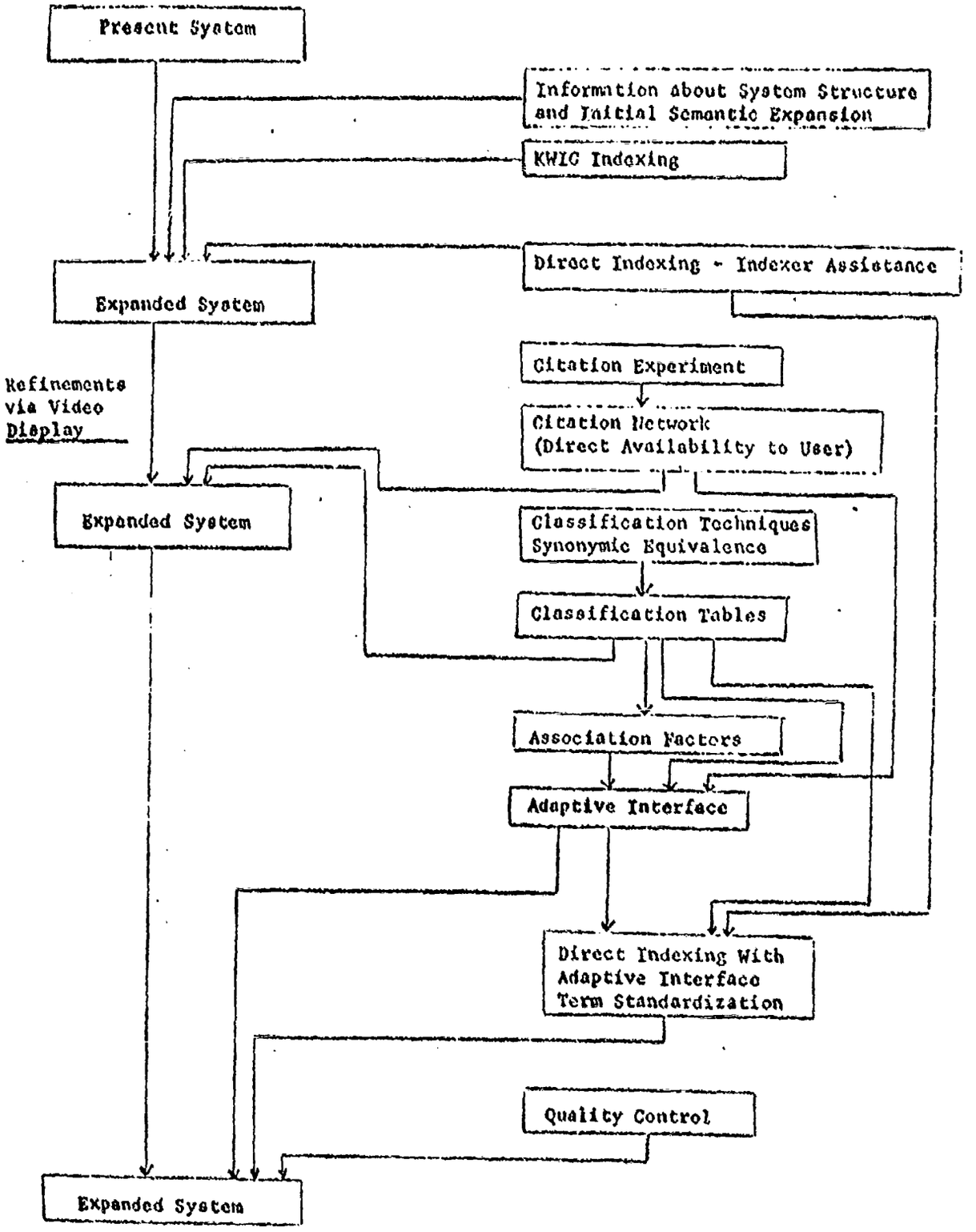


Figure 7 - Organization of Tasks

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
University of Pennsylvania
The Moore School of Electrical Engineering
Philadelphia, Pennsylvania 19104

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
THE MOORE SCHOOL INFORMATION SYSTEMS LABORATORY

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Scientific Interim

5. AUTHOR(S) (First name, middle initial, last name)
Morris Rubinoff

6. REPORT DATE
May 1967

7a. TOTAL NO. OF PAGES
28

7b. NO. OF REFS

8a. CONTRACT OR GRANT NO.
AF 49 (638)-1421

b. PROJECT NO.
9769-01

c. 61445014

d. 681304

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT
Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES
TECH, OTHER

12. SPONSORING MILITARY ACTIVITY
Air Force Office of Scientific Research
Directorate of Information Sciences (SRI)
Arlington, Virginia 22209

13. ABSTRACT
The report briefly describes the goals of the Moore School Information Laboratory which are primarily in the following tasks at the system level: specifications for machine storage of indexer aids, including lists of index items, synonymic equivalences, classification tables, and other semantic tools; organization of disk file storage; study of uses of graphic display. The report also summates the progress made by the Laboratory in these tasks.

DD FORM 1 NOV 65 1473

Security Classification