DOCUMENT RESUME

ED 087 454 IR 000 198

AUTHOR McIsaac, Donald N.; Olson, Thomas
TITLE Retrieval of ERIC Files. An On-Line Approach.
INSTITUTION Wisconsin Univ., Madison.
PUB DATE Apr 73
NOTE 10p.; Paper presented at the Association for Educational Data Systems Annual Convention (New Orleans, Louisiana, April 16 through 19, 1973)

EDRS PRICE MF-$0.65 HC-$3.29
DESCRIPTORS *Computer Programs; Computer Science; *Data Bases; *Information Retrieval; Information Storage; *Information Systems; *On Line Systems; Program Descriptions
IDENTIFIERS AEDS; Association for Educational Data Systems; Boolean Operators; Educational Resources Information Center; ERIC; Hash Coding; *WISE ONE

ABSTRACT

A description is provided for WISE-ONE, an information retrieval program designed to provide fast, efficient access to computer-based information files. The author focuses upon WISE-ONE's application to the Educational Resources Information Center's (ERIC) data base; WISE-ONE was designed specifically to meet the needs of researchers using ERIC, but its logic is sufficiently general to accommodate other computer-based library systems. The author first reviews ERIC's history, its products, and its thesaurus. Following this, the binary tree and index sequential file approaches to data structuring are discussed, and their disadvantages brought to light. Next, the hash coding method of data base entry is described, the conclusion is reached that it provides the best approach to the data base. Finally, the use of Boolean Operators is discussed and means of updating files are considered. (PB)

RETRIEVAL OF ERIC FILES

AN ON-LINE APPROACH

Dr. Donald N. McIsaac

Thomas Olson

University of Wisconsin - Madison

INTRODUCTION

The University of Wisconsin, like many other universities, holds the concept of research in education as precious. The researchers on the university campus need access to information from a variety of data bases in order to ensure that proper coverage of a given topic is complete and accurate. In order to be useful, a system designed to facilitate research must be responsive, inexpensive and complete. The development of the system described in this paper is a response to that need.

The computer, because of its ability to process at a high rate of speed provides a suitable medium for the processing of bibliographic information. This fact has been recognized by the leaders in many disciplines. Naturally, it is necessary to provide information in machine readable form if the computer is to provide assistance.

While there are computer tape services available from a wide variety of sources embracing many areas of interest, this paper will focus on the use of one, Educational Resources Information Center (ERIC). The principles of data storage and the retrieval mechanism discussed later in this paper are constant for all. The tape files available from the American Geological Institute, American Institute of Physics, American Society for Metals, etc., all follow similar patterns of development and distribution. Each of the many can provide a source data base for the WISE-ONE Retrieval program. A complete, but slightly dated, list of available tape services is discussed by Carroll (1970, ED044165).

The History of ERIC

The development of the ERIC idea began in 1959 with a federally supported study of an information system for new media growing out of NDEA. The number of federal documents reporting the research in education was growing at an incredible rate. Some mechanism for recording the results of research studies was badly needed. The initial study was to result in the ERIC idea.

DR. DONALD N. MCISAAC presently serves at the University of Wisconsin-Madison as Associate Dean, School of Education and Professor of Educational Administration. He is the past Director of the Wisconsin Information Systems for Education. He is a graduate of Pamora College (BA) and the Chasemont Graduate School (MA & Ph.D.).

THOMAS OLSON is presently employed as a specialist in data processing by the School of Education, University of Wisconsin. His responsibilities include the development of computer systems and programs for use in the School of Education. He earned his B.A. degree from the University of Wisconsin.

Western Reserve University contracted to develop a thesaurus of terms useful for indexing the rapidly accumulating research reports in the Office of Education. The thesaurus is a helpful document for manual use of the ERIC system in that each of the terms used in the indexing process is listed and related to other terms. The thesaurus becomes an essential document for the support of computer searches because of the need for accurate and complete descriptors.

The Office of Education looked across the Federal information systems for a model of systematic information collection. The National Aeronautic and Space Administration, the Atomic Energy Commission and the Department of Defense all provide government-based information services within their own area of concern. Each of these reflected a highly centralized approach appropriate to their method of organization. The centralization which made these efforts effective were not viewed as reasonable for the ERIC concept.

A decentralized approach which based upon the review of documents by subject specialists rather than information specialists was developed. The field of Education was divided into several basic categories. ERIC clearinghouses were established and began to process documents in a variety of educational areas of concern.

I. Person centered clearinghouses

Adult Education

Disadvantaged Education

Early Childhood Education

Exceptional Children

Rural Education

Teacher Education

II. Subject or Skill Centered Clearinghouses

Linguistics

English

Science

Reading

Foreign Language

Vocational Technical

III. Functional Centered Clearinghouses

Counseling

Educational Administration

Library and Information Sciences

IV. Level Centered Clearinghouses

Junior Colleges

Higher Education

V. Other Clearinghouses

Media and Technology

Tests, Measurement and Evaluation

It was decided that the reproduction and computer services would be contracted. The actual indexing and abstracting began with the formal creation of ERIC in 1964. The opportunity grew out of the ESEA Title I in conjunction with the development of the document collection on the disadvantaged. In 1965, Bell and Howell contracted for the document reproduction. This process was subsequently assumed by NCR. In 1966, the contract for storage and retrieval processing was let. This significant step provided the basis for providing the ERIC data in machine useable form.

By Fall of 1966, a dozen clearinghouses were established and so most of the elements of the system were in place. Each of the clearinghouses was staffed to review and abstract the documents published as the result of federally supported research. The abstract concept was broadened to include speeches, technical papers, occasional papers, conference proceedings and the like. By doing so, the problem of quality control was amplified. Each of the clearinghouses held the responsibility for selecting the information to be preserved. The ERIC Central maintained the responsibility for reproduction of both the paper, microfisch, and computer materials through contracted services.

ERIC PRODUCTS

Several products have emerged from the ERIC idea and each is designed to make the system a more helpful research tool for its users. Research in Education (RIE) is a monthly abstract journal reporting recently completed research reports, descriptions of outstanding programs, and other documents of educational significance. The report also includes a section on the newly funded research projects supported by the Office of Education. The RIE is indexed by subject, author or investigator, and institution. The publication is designed to provide a basis for continuing announcement of publication availability and to provide a basis for comprehensive current awareness. ERIC also supplies a semi-annual and an annual cumulative index.

The research documents available through ERIC only cover a small portion of the literature. The clearinghouses were established to review what might be considered the fugitive literature. The Current Index to Journals in Education is a manual guide to the periodical literature with coverage of more than 500 major educational and education related publications. The CIJE includes a main entry section with annotations and is indexed by subject and author. Demiannual and annual cumulative indexes are also available.

Several special collections are also prepared for special subject searches. These include such collections as:

Office of Education Research Reports

Pacesetters in Innovation

Manpower Research

ERIC Catalog of Selected Documents on the Disadvantaged

Selected Documents in Higher Education

The ERIC thesaurus is an essential document for the access of information in the ERIC system. The RIE and CIJE provide access to documents within a given time-frame. But generally, one is interested in a literature review covering a broader time-span than one month or one year. Coordinating the manual search for documents covering a long period of development is difficult. This will become increasingly true as the ERIC files continue to grow.

THE ERIC THESAURUS

The original thesaurus grew out of the work performed at Western Reserve University. Continued development has continued as the result of the delicate interaction between the ERIC Clearinghouses and ERIC central. As new areas of concern creep into the educational literature appropriate descriptors may be suggested to ERIC central. The interplay between the abstractors and the lexicographers of ERIC central filters out the new terms which are to be added as descriptors. This verbal tug of war between the decentralized clearinghouses and the centralized ERIC central serves to sharpen the language of education. It produces a kind of authority list of words in education formulated within the format of a thesaurus. The thesaurus is the document which serves as the guide for identifying given references. The descriptors represent the coordinates employed to locate a labeled reference. For once a document has been entered into the system, it is immediately buried, and may only be uncovered by proper identification of its descriptive coordinates.

The thesaurus offers a list of descriptors which serve as labels for documents within the system. These are listed in alphabetical order enabling relatively simple access to appropriate entry points. The system was designed to assist the user in search of the correct descriptor. This is accomplished by listing related terms, broader terms, narrower terms, and synonymous terms along with the descriptor. The user of the thesaurus is directed to additional terms which may be useful for widening or narrowing his search pattern. The following diagram illustrates the relationship of sub-terms associated with a given descriptor:

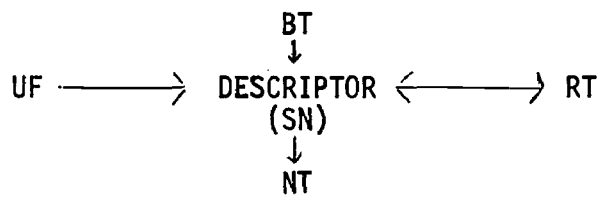

Figure 1
Diagram of Thesaurus Notation

The narrower term (NT) listed under the descriptor suggests alternative descriptors which would serve to narrow the search. The narrower term denotes a hierarhical relationship between the main descriptor entry and a descriptor

which belongs to the same class but is on a lower level of the hierarchy. The narrower term is more specific.

"Broader term" (BT) is the second hierarchical notation and is the reciprocal of the "narrower term" notation. A BT entry indicates that the descriptor is of the same class as the main descriptor entry but that it is on a higher level of the hierarchy.

"Related term" (RT) performs a dual function. It serves to clarify the scope of the main entry by describing the context in which the main descriptor which may be of use to the user.

"Scope notes" (SN) appear in the thesaurus to clarify the meaning of a particular descriptor is ambiguous, these scope notes are useful.

"Used for" (UF) is an additional notation in the thesaurus specifying that the main descriptor is viewed as a synonym to the "used for" label. Words specified in this manner do not appear in the index and a search on the term will be useless. One additional notation, "use," (U) specifies the term as one which does not appear in the index and the user must enter the system under the term which follows the (U) specification.

The thesaurus is the key for entering a computer-based search of the ERIC files. It is necessary for any user of a computer-based search to be familiar with the contents of the thesaurus. For with an understanding of the elements and structure of the thesaurus, he is better able to construct intelligent search formulas.

Since its inception, ERIC has enjoyed a healthy growth. The addition of Report Resumes (RIE-ED Series) has shown a gradual increase to a cumulative total of 59,575 entries in the beginning of 1973. The rate of accession has also been increasing and now approximately 12,000 new documents are added each year. The Journal file has enjoyed an even greater growth. Since its beginning in 1968, the Journal Article Resumes (CIJE-EJ Series) has ballooned to 62,751 document references by January 1973. The current rate of growth of this file is 17,000 references per year. The WISE-ONE Computer-Based Retrieval System.

WISE-ONE is designed to provide fast, and efficient access to computer-based information files. It was designed specifically to meet the needs of researchers requiring access to the ERIC system, but the logic is sufficiently general to accomodate any of the many computer-based library systems. In order to provide for the on-line computer access capability, it is necessary to restructure the data-base. In the remaining sections of this paper, a variety of alternatives are discussed and the WISE-ONE system and logic is explained.

THE BINARY TREE

One possible data structure is a binary tree which is often employed for processing natural language and computer compilers. The binary tree relies upon a carefully contrived data base where entry is always at a common point followed by selective branching until the correct key is found. The easiest way to conceptualize the tree structure is to trace the creation of the data base because the search and creation logic are identical. Let's start with a coded word. It is stored with the associated data record-awaiting comparison with the second keyword. A numeric compare with the second keyword will produce a negative, positive or zero result. When the result is equality, the

data record is expanded with the text associated with the new keyword. A negative result will cause a left node pointer to be selected from an available space list. The keyword and associated data record is then stored in the location specified by that pointer. Similarly, a right-node position is selected when the compare result is positive. Thus, two pointers may be identified with each node or keyword in the data structure. As each new keyword and data record is entered, it follows the logical path of left or right node pointers until an empty node is encountered. The data record and associated pointer is then added to the tree. When a search is desired, the search follows a path through the node pointers until a compare on the keyword produces a zero result. The data record associated with that node is the desired information. The path is dictated by plus or minus results for a compare between the search keyword and the node keyword.

Several advantages and disadvantages accrue from this method. Dynamic creation of the data base is possible. Updates to the system are quite simple. Deletions are not easily accomodated as they interrupt the flow of the tree. A balanced tree, while not essential, provides a more efficient search. The logic for balancing a tree structure is extremely complex. Search trees are relatively short but increase as a log funtion of data base size.

The major drawback to this structure occurs when the structure is being updated or corrected. To keep the search times optional, it is important that the tree be fully balanced, such that all the nodes on one level are filled before the nodes on the next level are used. If this balancing is not done, the tree becomes so unbalanced that search time is significantly degenerated. Algorithms to balance the tree structure exist but they are inefficient when working on mass storage because they require a large number of I/O requests. Another problem is the high overhead associated with placing new information into the file. For example, to insert a new node on a fully balanced tree of 16,000 nodes requires 16 I/O requests before the proper parent is found. This overhead becomes substantial when a large number of new nodes are to be added or the existing tree is unbalanced. In addition, a large amount of space is consumed by the linkage information when a large number of nodes are stored in the tree. The problems associated with the tree structure indicate it is only a highly desirable procedure when used with small or static data bases.

INDEX SEQUENTIAL FILES

A second approach is commonly called index sequential. In this frequently employed method, the file is sequentially ordered and an index of references is developed from the file. Thus, a search may be limited to the index, locating the approximate search entry point into the main file. An index sequential approach cuts the search-time significantly from a sequential search by locating specific search entry points and eliminating the need to examine each record. However, large files require large indexes and the method only delays the need to consider more efficient procedures. It is simple to employ and therefore is highly desirable when operating with relatively small files or when the search response time is not critical. Updates are relatively simple because of the sequential order of the file. Update information may be merged into the sequential file. This does require that the file be recopied, which may be a costly method when the file is very large. Many variations have been developed in the interest of optimizing the updat procedures of index sequential files.

However, for the application to an on-line inquiry to a large bibliographic data base, the search time is generally not sufficiently responsive. For this reason, the WISE-ONE staff settled on a hash coding scheme for citation identification.

HASH CODING FOR DATA BASE ENTRY

The hashing method may be employed as a variation of the Index Sequential Approach in which the index is hashed using a mathematical permutation of the keys to determine the approximate location of the citation in the file. This method is efficient but may lead to slow response time when employed in an interactive mode.

The structure of the WISE-ONE data-base is a linked table scheme and is an adaptation of a direct chain, hashing scheme employing a linked list structure.* There are three types of tables developed in this process, a base-table, collision tables and citation tables. The heart of the system is the hash coding scheme which is incorporated into the data-base structure. A hash code is a method of computing the storage location of a record based on some mathematical permutation of the search key. The hash algorithm WISE-ONE employs generates two numbers: the hash address and the virtual key or residue. These two numbers correspond to the remainder and quotient of the division of the keyword bit pattern by the size of the base table. The role of the hashing scheme and the collision tables in the structure of the data-base is best explained by tracing the search process. See Figure 2.

When a search key is entered, it is hashed by multiplying successive sets of six character computer words until the entire keyword is stored as a 72 bit product. The middle 13 bits are selected as the hash address for entry into the collision table. This hashing approach is an approximation of the middle squares approach. The method produces a random bit pattern, therefore, reducing the probability of collisions. It is obvious that at some time two or more keywords may result in the same hash address. The collision table is designed to resolve these conflicts. When the hash address is computed, the surrounding 36 bits are selected as the residue.

The collision tables are too large to store in memory. Therefore, we need a mechanism to convert the hash code to a mass storage address for the collision table. The hash address is used to point into a base table. The base table is a core resident list which contains the address of all collision tables which reside on secondary storage.

The collision table is entered at the hash address and the stored residues are compared. The residue is stored as a psuedo keyword in the interest of storage efficiency. An equality compare on the residue associated with a given hash address points to a citation table in which all references to the given key word are stored. These references constitute a search queue which may interact through boolean operators with a prior search queue to produce a resultant list. The process is repeated with additional keywords until the search logic is completed.

---

*The ERIC search program - WISE-ONE - was funded by the School of Education, Department of Educational Administration, Wisconsin Information Systems for Education (WISE. Mr. S. C. Yang and Professor Venesky contributed to the development of the hashing scheme. The program was also a class project in Computer Science - CS 638 taught by Professor Travis. These contributions are acknowledged and appreciated.

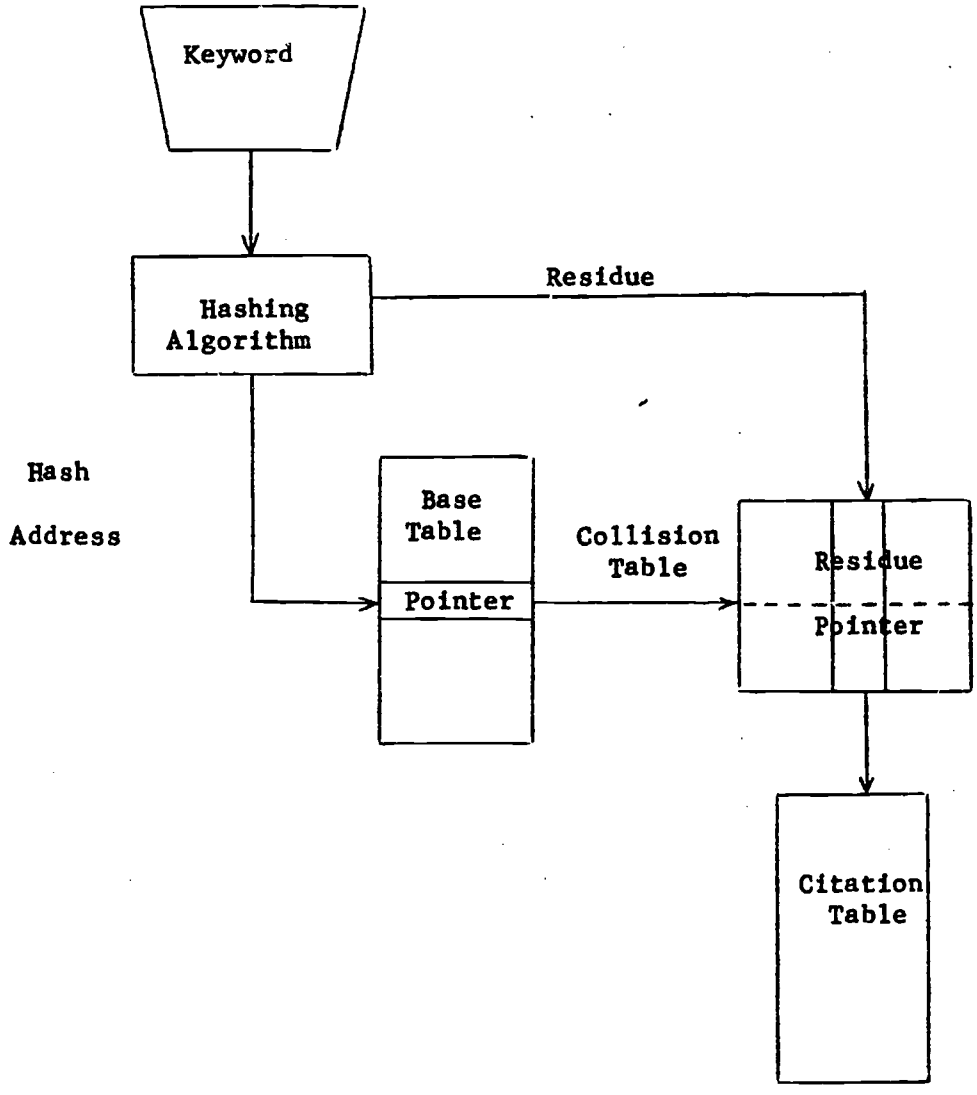

FIGURE 2

Hash Coding Scheme

When the constructed search queue is complete, the citation numbers are hashed in the same manner as the keywords. The hash address is used as a pointer to another base-table to obtain a link to a collision table on secondary storage. The collision table is then searched for the matching virtual key or residue and the associated link is then followed to obtain the title, author and journal citation of the ERIC number. This process is repeated for each citation number in the search queue until all citations have been printed.

BOOLEAN OPERATORS

It is convenient to think of three types of queues. The secondary queue is the result of a keyword search. It contains a list of the citations associated with a given keyword. The primary queue is a prior list of citations which interact with the secondary queue through a boolean operator. It may be empty. The resultant queue is that which is produced by the interaction of the primary and secondary queues through a boolean operator.

The boolean operators for WISE-ONE include AND, OR, and NAND. The AND operator generates the resultant queue which contains citations common to both the secondary and primary queues.

Primary Queue		Secondary Queue		Resultant Queue
Cit. 1		Cit. 2		Cit. 6
Cit. 3	.AND.	Cit. 6	=	Cit. 9
Cit. 6		Cit. 8		
Cit. 9				

The OR operator generates a resultant queue which contains citations unique to both the primary and secondary queue. The OR operator is therefore additive in nature.

Primary Queue		Secondary Queue		Resultant Queue
Cit. 1		Cit. 2		Cit. 1
Cit. 3	.OR.	Cit. 3	=	Cit. 2
		Cit. 4		Cit. 3
				Cit. 4

The NAND operator is a BUTNOT operator which reduces the primary queue by all matches within the secondary queue. It is helpful for systematic reduction of the primary queue.

Primary Queue		Secondary Queue		Resultant Queue
Cit. 1		Cit. 1		
Cit. 2		Cit. 3		Cit. 2
Cit. 3	.NAND	Cit. 5	=	Cit. 4
Cit. 4				
Cit. 5				

The boolean operators permit the dynamic construction of search formulas. Each keyword entry will involve the hash algorithm and will produce a resultant queue as prescribed by the selected operator. In order to optimize the building of search formulas, it is useful to employ parenthetic logic.

(COLLEGES.OR. UNIVERSITIES.OR.HIGHER EDUCATION).AND.(FISCAL SUPPORT.OR.FINANCE)

This approach expands the utility of searching large and complex data bases. The nature of the ERIC files requires that such a capability be available.

UPDATING THE FILE

The creation and update of the data-base follow a different line of development than the search process. The keywords in the form of descriptors, identifiers and author's last names are abstracted from the ERIC tapes along with the title, author and date of the citation. Each keyword is hashed and the hash address residue and keyword are written into a file along with the ERIC citation number. The title, author and citation numbers are written into another file. The keyword file is then sorted on citation number within residue within hash address. This file is then merged with the existing master file to create a new master file. The master file contains all the information in the proper order for easy generation of the table structure. The data-base search files are then generated from master file and the title and author file.

There are a number of advantages to this method of storage and retrieval, the most notable being its extremely fast search time. The CPU time* per keyword is in the order of hundreths of seconds. The overall search time is less than a tenth of a second per keyword.

Another important feature of this search method is that search time will not increase significantly as the data-base grows in size. This is because the number of probes to the disk to search for any keyword is two, one to read the collision table and one to read the citation table. The only portions of search-time that will increase are those associated with the collision table search for the residue and the time required for boolean process of the longer lists.

---

*WISE-ONE currently runs on the Univac 1108 at the computing center on the University of Wisconsin-Madison campus. It is written in 1108 assembler and Fortran V. It uses about 31k 36 bit words of core storage and about 1500k words of disk storage for each file. This can be translated to 124k bytes of memory and 6 megabytes of mass storage on IBM Systems. The nature of the hashing scheme forces the code to be machine dependent and it would require considerable reprogramming to run this system on computers other than UNIVAC 1100 series machines.