may be searched and retrieved using similar retrieval programs and queries for all of the sources.

2. By developing a center-oriented file format, it is possible to combine records originating in the large-scale external data sources with locally generated data records.[2] These data records, which may be highly important for the members of the local scientific community, may be distributed through the system and used commonly among the local users.

In generating a sequential file consisting of variable length logical records, it may sometimes be better, or necessary, to choose a maximum blocklength which is smaller than the maximum record length. The present block structure for STF should generally be applicable in the above situation; it will certainly be convenient when most logical records are expected to be much shorter than the maximum blocklength.

ACKNOWLEDGMENT

The first step of the operation was made possible with the intellectual, technical, and administrative help of Satoru Hoshino, Ichiro Nakagawa, Yasuyuki Sakamoto, and several other members of the Data Processing Center, Kyoto University. The SDF tapes were purchased with funds given by the Ministry of Education of Japan, Tokutei Kenkyu I, Showa 45, No. 99042 and Showa 46, No. 99032.

LITERATURE CITED

(1) Yamamoto, T., and S. Fujiwara, "Syntactical Proximity—Partial Syntactical Analysis of Natural Language Data Records," *J. Chem. Doc.* **11,** 256-7 (1971).

(2) Yamamoto, T., T. Kumai, K. Nakano, C. Ikeda, T. L. Kunii, H. Takahasi, and S. Fujiwara, "Todai Scientific Information Retrieval (TSIR-1) System. I. Generation, Updating and Listing of a Scientific Literature Data Base by Conversational Input," *J. Chem. Doc.* **11,** 228-31 (1971).

(3) Yamamoto, T., K. Nakano, C. Ikeda, T. L. Kunii, H. Takahasi, and S. Fujiwara, "On-Line Tape IOCS for an Information Retrieval System," unpublished work.

(4) Anzelmo, F. D., "A Data Storage Format for Information System Files," *IEEE Trans. Computers* **C-20**(1), 39 (1971).

(5) Chemical Abstracts Service, "Standard Distribution Format Technical Specifications," Columbus, Ohio, 1970.

(6) Chemical Abstracts Service, "Data Content Specifications for CA Condensates in Standard Distribution Format," Columbus, Ohio, 1970.

(7) Chemical Abstracts Service, "Samples of Chemical Abstracts Condensates Data Records in Standard Distribution Format," Columbus, Ohio, 1971.

Du Pont Information Flow System*

WARREN S. HOFFMAN
Information Systems Division, Secretary's Department, E. I. du Pont De Nemours & Co., Inc. Wilmington, Del. 19898

Received February 23, 1972

The Information Flow System is a large-scale information retrieval system developed for processing of Du Pont information files. As currently implemented, the system stores and retrieves information on company technical reports. Important features of the system include the use of threaded lists in addition to inverted files to permit optimum searching. Users prepare searches in a free format query language, which is then optimized by the system to make most efficient use of the file structure. Answers are in the form of accession numbers or abstracts. Extensions of the system for handling chemical structure information and on-line processing are also discussed.

In 1964, Du Pont established a centralized group for indexing and searching company technical reports. Mainly to service this operation, an information retrieval system was developed using the IBM 1410 computer. The system was originally designed as an interim system, which would have an anticipated lifetime of less than five years and would be replaced by a more modern system when file size and economics dictated. The application of this system in Du Pont's Central Report Index has been discussed.[1] In recent years, the volume of input and searches using this system has steadily increased, and the file size has grown substantially.

In late 1966, a study was undertaken to determine future machine processing requirements of the report group. The objective of the study was to design an evolving system to handle increasing work loads and new services more efficiently and in a shorter cycle time. The new system was to make use of modern hardware and software concepts to provide a higher level of service than available in the past. The Information Flow System (IFS), which was started up during 1971, is essentially the product envisaged during the initial study.

Between 1966 and 1971, a detailed file organization scheme was developed. The design and proposed system facilities were reviewed with the intended users. A basic data report specifying functional attributes of the proposed system was prepared, followed by detailed design and implementation. Conversion programs to generate the master files were also written.

*Presented before Division of Chemical Documentation, Mid-Atlantic Regional Meeting, ACS, Philadelphia, Pa., Feb. 15, 1972.

UTILIZATION

The Information Flow System is used primarily to store information extracted from Du Pont Technical Reports. The Central Report Index employs professional scientists and engineers to prepare input and searches. The input process consists of selecting keywords from a controlled vocabulary which adequately describe the document for purposes of later retrieval. Terms are intellectually divided into two categories: Chemical Terms, which refer to specific chemical compounds, such as benzene, ethyl alcohol, polymers, or generic chemical concepts; and General Terms, which refer to other aspects of the concepts discussed in the report. Chemical names are indexed with roles to indicate the context in which the term was used in the document. For example, roles can distinguish between discussion of the substance as a product *vs.* a reactant *vs.* a solvent, etc. Reports are indexed using links to distinguish completely separate concepts mentioned in one document.

This paper concentrates on the computer system used to enter and search the keywords associated with each document. Associated systems will be mentioned briefly.

OBJECTIVES

From a systems viewpoint, information retrieval systems do not fall neatly into the categories of commercial programs, scientific programs, or systems software. Search systems bear some resemblance to systems software, in that the tasks the system actually performs are determined only when queries are submitted. This is in contrast to both commercial and scientific programs, which are generally oriented to a specific task. Information retrieval systems deal with data bases, which typically have two unusual attributes: much more information is added than is deleted, and search activity is higher than update activity. Many statistics cite the rapid growth of information in recent years. The task of information retrieval system developers and operators, and particularly the problem addressed by the Information Flow System, is how to maintain and search an ever-growing data base without drastically increasing costs. While substantial help in this direction has been achieved by the general downward trend in computer costs with time, growth in file size and utilization has made other solutions necessary. The goal of the Information Flow System was to be able to handle increasing work loads less expensively by utilizing the most advanced hardware and software concepts available.

After a preliminary analysis, it appeared that the two directions holding the greatest promise for achieving the foregoing objectives were minimizing overhead owing to file size, and developing a scheme by which only a minimum portion of the data base would be manipulated during both updating and searching. Using direct access files and a highly indexed data base were judged to be necessary. Direct access files are files stored on media from which records can be retrieved in a single operation without serial processing of large parts of the file, usually by specifying the record address and an identifying key. Most existing direct access files, including those of the Information Flow System, utilize magnetic disks. Other types of direct access devices include magnetic drums and magnetic strips.

Most existing systems, including the predecessor system to the Information Flow System (IFS), used magnetic tape as storage media, even when inverted lists were employed. As a result, a large cost penalty was incurred for processing master file tapes to reach the desired information before any logical operations could be performed. In a direct access environment, any portion of the master file can be rapidly retrieved. With the advent of removable disk devices, direct access information retrieval systems became economically feasible.

To minimize the percentage of the data base that would have to be examined during update and search operations, two concepts are employed. First, documents are divided into two categories: those which have all input processing completed, and those which are still in a state of flux. For the former, a more compact storage format is used, while additional infortion is retained with the latter to facilitate updating. New records are assigned to available locations within the master file. The vast majority of existing documents are not disturbed during update. Instead, indexes which are of more modest size are altered to include the new information.

Second, to permit examination of the minimum portion of the data base when searching, a threaded or linked list form of file organization is used. Lefkovitz[2] has given an excellent discussion of file processing techniques employing linked lists. Many of the IFS concepts are based on the concepts discussed. The file contains a single record for each document, storing all terms by which the document is indexed. Associated with each term is the record address of the next document in the file which also references that term. Files are processed by examining each record in turn, a process referred to as "threading." Using this scheme, only a small fraction of the master file records is accessed in the typical case. Because new terms with few references are constantly generated as new documents are added to the system, searches do not have to examine a substantially increased number of records, regardless of file size. Hence, the goal of minimizing the effect of growth in file size on system economics is furthered.

Statistical listings indicate that some common terms, such as "water," reference a significant percentage of the documents in the collection. For these terms, which number less than one-half of one per cent of the total vocabulary, inverted lists are maintained instead of threaded lists. In an inverted list, the identities of all documents which reference the term are collected and stored in records, reducing the number of record accesses required to obtain that information. In summary, the Information Flow System file structure can be described as a combined inverted and threaded list scheme. Terms with a small number of postings, known as Minor Terms, are maintained using threaded lists. Terms with a large number of postings, known as Major Terms, are maintained using inverted lists.

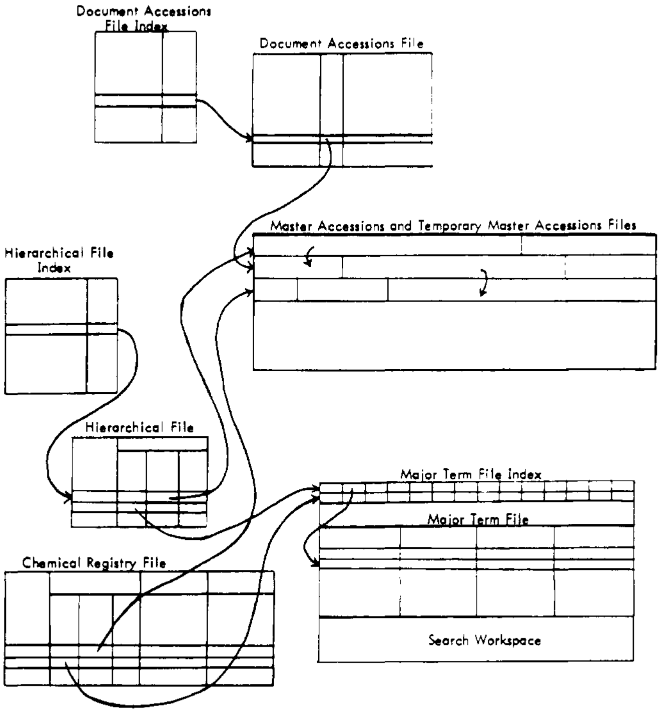
SYSTEM IMPLEMENTATION

File Structure. The Information Flow System file structure consists of nine major files, all of which reside on magnetic disks. These files fall into four categories, which will be discussed separately.

Category 1. Files which store information about each document in the system

Category 2. Files which store the terms by which each document was indexed.

Category 3. Files which store information about the terms which are valid in the system.



Note: Detailed information on file formats is shown in Figures 2, 3, and 4

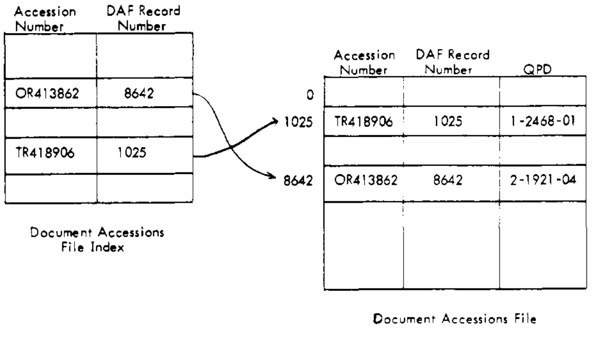
Figure 1. Information flow system file structure

Category 4. Files which provide access to inverted lists for terms which have many references in the files.

An over-all picture of the file structure is shown in Figure 1.

Document Related Files. Category 1 contains the Document Accessions File (DAF) and the Document Accessions File Index (DAF Index) as shown in Figure 2. The DAF is a direct access file which contains an entry for every document in the system. When a new document is added, the next available location is assigned. This position number is thereafter used as the internal accession number of the document within the IFS files. The DAF Index is an indexed sequential file which permits access to DAF entries by external accession number during updating and for generating selected printouts of records for maintenance purposes. All valid external accession numbers are stored in the DAF Index.

The DAF contains two important entries: the external document accession number, and the disk address of the record which holds the indexing information for the document. The external accession number is maintained for use during searching when it is necessary to convert the internal accession number for each answer back to the document number as known to the user. The disk address of the record containing the indexed terms, whose format and content will be described under Category 2, consists of three parts: the disk volume number, track number, and record number on the track. As the files were expected to be quite large, provision was made for up to nine separately addressed disk volumes or data sets. Each master file record is assigned to a track within one of these data sets. Tracks are packed as full as possible. The record number on the track is used as a key to distinguish among the various documents. The three fields are referred to collectively as the Qualified Position Designator (QPD). Provision has been made for adding additional fields to the DAF record to store either actual information or the address of information in additional related files. For example, alternate document identification numbers, dates of issue, authors, or other bibliographic or textual information could be referenced. The primary reason for storing the indexing information separately was to restrict the DAF record to fixed length information.



Accession Number	DAF Record Number
OR413862	8642
TR418906	1025

Document Accessions File Index

	Accession Number	DAF Record Number	QPD
0			
1025	TR418906	1025	1-2468-01
8642	OR413862	8642	2-1921-04

Document Accessions File

Figure 2. Category 1 files. Document accessions file and document accessions file index

Category 2 files consist of one or more Master Accessions Files (MAF) and a Temporary Master Accessions File (TMAF). One or more records exist for each document, the set containing all terms by which the document was indexed. While the size of the data base made it necessary to provide for more than one MAF, these files logically form a unit, and are addressed identically by all programs. The TMAF is used to store documents added recently which may be undergoing a high rate of term addition and deletion. If links were used to index a document in the file, each MAF or TMAF record may contain as many links as can fit within the maximum size record selected. If more space is required, continuation records are used. Each record contains a field which contains the address of the next record, if any, in the continuation chain. Few documents require continuation records. Further discussion of these files will pertain to the organization within a single link field.

Each MAF and TMAF record is divided logically into two parts as shown in Figure 3. The first part, used to store major terms for which inverted lists are maintained elsewhere, contains the internal term codes for each major term in ascending order. Following this is a similar series of entries for each minor term, for which threaded lists are maintained. The entry for each major or minor term consists of two parts, a term code, and a role field. The role field contains the role offset and number of role segments. The role offset field contains the location, relative to the beginning of the link, where the roles used to index the term in the given document are stored. The number of roles field stores the number of different roles used. If no roles were used to index the term, this latter field contains zero.

For major terms, as shown in Figure 3, the role offset field points to a location where roles are stored in a one-role-per-character format. For minor terms, however, the role offset field points to a location where the role is followed by the QPD of the next document in the file which references that term in the specified role. These entries are repeated for as many roles as were used to index the term. Immediately preceding the first role entry is the QPD of the next document which referenced the term in any role. This additional information is stored for use when searches are made for terms in multiple roles, or

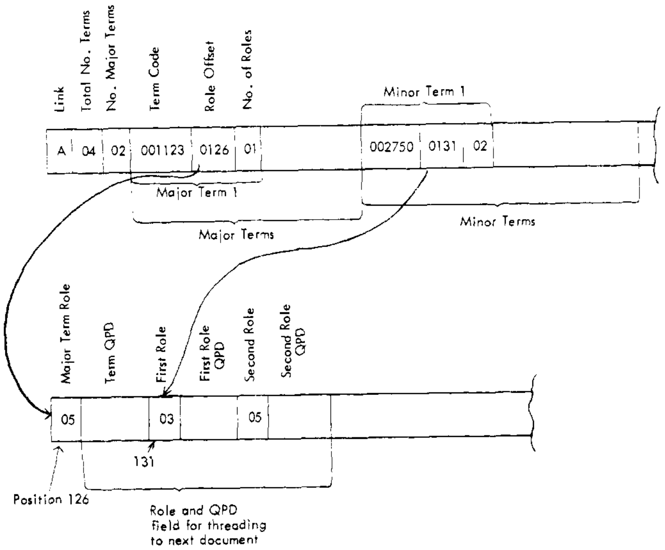
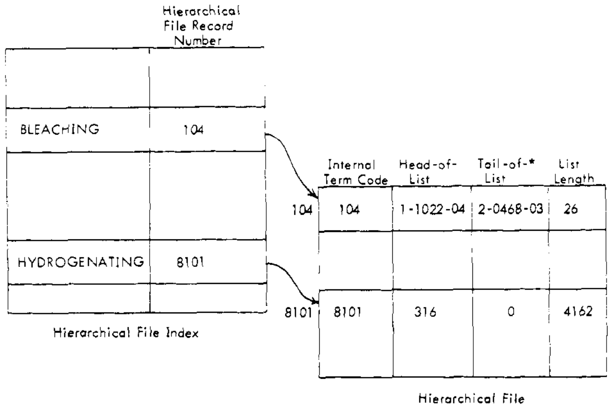


Figure 3. Category 2 files. Master accessions file format



*Zero indicates term is major. Major Term File Index record location is stored in Head-of-List field.

Figure 4. Category 3 files. Hierarchical file and hierarchical file index

when a term was indexed without roles. When a search is made for a term in a single role, the QPD associated with that role is employed, thereby achieving a more economical search. An additional reason for use of this scheme will be discussed in connection with Category 3 files below.

The Temporary Master Accessions File records are identical in format to the Master Accessions File records, except that for minor terms, the address of the previous document which referenced the term in a specific role, or without regard to role, is also stored. This is of use during updating. At the front of both MAF and TMAF records, the internal document accession number of the document is stored to permit association with the external accession number. Documents are moved from the TMAF to an MAF periodically when all processing is complete.

Term Related Files. Category 3 files include the Hierarchical File (HF), Hierarchical File Index (both shown in Figure 4), and the Chemical Registry File (CRF). These files store information concerning the general and chemical terms which are valid in the system. The HF and CRF are analogous in function at the term level to the DAF at the document level. The Hierarchical File contains an entry for each general term. New terms are assigned to the next available location, and this location serves as the internal number by which the general term is known to the system. As users desire to refer to the term by an alphanumeric representation, the Hierarchical File Index, an indexed sequential file, is used to store the internal number for each general term. General terms can contain up to 35 characters.

The Hierarchical File contains three main entries for each term: the head-of-list, tail-of-list, and list length. The head-of-list field contains the QPD of the first document which contains a reference to the term. During searching operations, this field is accessed to initiate a search for all documents relating to a term specified in a search. The tail-of-list field stores the QPD of the last document which references the specified minor term. This field is only used during updating. The list length field contains a count of the number of documents which are referenced by the term, and is primarily used for optimization of search execution logic. The algebraic sign of the list length field indicates whether a term must or must not have roles. These data are established when the term is added to the system. The head-of-list QPD refers to the first document which references the term regardless of role. During searching, documents on this thread are examined in turn until the first document which references the term in the role specified during the search is located. If the search is for a single role, the QPD associated with the role in the MAF or TMAF record is used to access only documents which reference the term in the specified role. By this mechanism, only one head-of-list QPD need be maintained, rather than a separate head-of-list QPD for each role. For major terms, the head-of-list QPD field is used to contain the address of a record in the Category 4 Major Term File Index described below to process the inverted lists in the Major Term File. Major and minor terms may appear in any location in the HF.

The Chemical Registry File performs an analogous function to the Hierarchical File for chemical terms. The major difference between the two is that the CRF is addressed directly using the compound number as a key to the position within the file where the information for the specific compound is stored. This is possible because numbers are used as chemical term codes. Approximately 275,000 numbers are used in five distinct ranges. A number is subtracted from the compound number based on which range it is in to generate a key within the 275,000 valid positions. In addition, the CRF contains multiple sets of head-of-list, tail-of-list, and list length fields, while the HF contains only one of each field. These separate sets contain the corresponding information for several potential users of the IFS. Because the same number would be used to identify the identical compound by each group, a common CRF is used. However, when the system is operated for a specific user, only one of the sets within the CRF is addressed. No other IFS files are shared in this manner.

Inverted Files. Category 4 files consist of inverted lists for major terms and work space for use by the search program. These lists are organized into fixed length records, with as many records being used to contain the postings as required. Each posting consists of a QPD-link-internal accession number triplet. A separate inverted list is maintained for each major term. Multiple lists are maintained for major terms which can have roles. To locate the beginning of the inverted list for each major term, one record is reserved for each major term in the Major Term

File Index. This logical file is physically located preceding the inverted lists in the Major Term File data set.

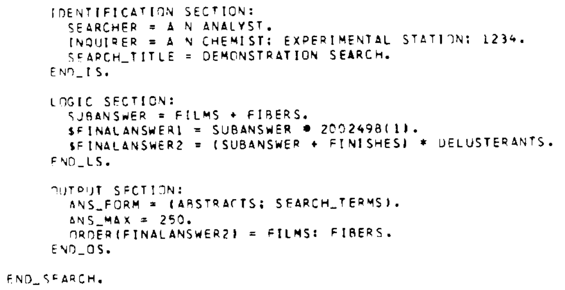
As discussed in the foregoing, the Hierarchical and Chemical Registry Files contain the address of the record in the Major Term File Index for each major term. This record contains a dedicated position for each role and an additional position for terms which do not have roles. These positions contain the number of postings in each inverted list, and a number which allows the location of the beginning of each inverted list to be calculated. The address is obtained by dividing the latter number by the number of postings stored per record. The quotient is the record number and the remainder the position within the record. This technique permits inverted lists to start and end within a record in the Major Term File, thereby reducing storage requirements. All records beyond the last posting for the last major term are used for work space for the search program, as discussed below.

Search Logic. During design and development of the Information Flow System, it was clear that substantial savings in searching would be required to overcome the anticipated higher costs of updating, relative to the IFS predecessor system. A great deal of analysis went into development of the search programs to meet the goals discussed in the introduction to this paper—namely, that of minimizing the effect of file size and permitting direct access to the relevant portions of the collection. In basic form, the search facilities offered by most keyword-oriented systems are similar. A Search Language Definition report was prepared and served as the basis of the implementation effort. A sample search composed with this language will be used to discuss the search system facilities (see Figure 5).

Information Flow System searches consist of three sections: the Identification Section serves to identify the user and contain certain optional items, the Logic Section controls the selection of master file records, and the Output Section defines formatting of search answers and generation of optional data. Like most retrieval systems, an IFS search consists of three phases. In the first, queries are checked for proper syntax and semantics and tables are constructed to control the remainder of execution; in the second, the search is executed obtaining as answers the internal accession number of documents which meet the specified criteria; and in the third, output is formatted. Special logic to perform generic searches will be discussed separately.

Search Construction. The Identification Section contains options which define the name of the person preparing the search, the name of the person for whom the search is being executed, and an output page heading. This section is mainly checked only for syntax; no record is maintained of the data submitted other than the heading. The Logic Section contains one or more statements which define criteria for considering a document to be an answer. Statements may be subanswers or final answers. The two types are coded identically, except that final answers are delineated by a dollar sign preceding the name. Each subanswer or final answer consists of one or more terms related by the standard Boolean operators: "And" represented by "*". "Or", represented by "+", and "Not" represented by "¬". Terms can be general or chemical terms, and may have roles appended to them in parentheses. Chemical terms must be identified by their registry number and not by name. The letter following the chemical number is a check digit used to verify that a valid compound is being coded. In both subanswers and final answers, previously-defined subanswers may be inserted as pseudo-terms. Subanswers are primarily used when logic would be common to two or more final answers. Each question must contain at least one final answer. Because query logic optimization is done at the subanswer or final answer level, searches are written as a single statement whenever possible.

A basic design philosophy of the IFS is that it should not be necessary for users to be intimately familiar with the file structure to obtain optimum results from the system. For example, users are not even aware of which terms are designated as major. It is the task of the Search Edit program to prepare information for later phases in such a way that the effort expended in searching the files will be at a minimum. This can be illustrated by the examples of statements shown in Figure 6. The letters A, B, and C represent terms, below which are listed a hypothetical number of postings, as stored in the Chemical Registry File or Hierarchical File. In Example 1, the search requests a simple intersection of terms A and B. Both terms are minor because of their short list lengths. The query analysis routine will decide to search down through the thread for term A, as it has the fewer postings of the two, and check each document reference for B. In Example 2, term A is a major term, owing to its large number of postings, while term B remains a minor term. In this instance, the search program will choose to thread down the list for term B, checking each document to see if it contains term A. In Example 3, both terms A and B are major. The search will compare items on the inverted lists. Answers will be those which appear on each list. In Example 4, term A is a major term, while terms B and C are minor terms. The inquiry analysis will decide that the threaded lists should be examined for terms B and C, and each document checked to see if it contains term A. Duplicates will be eliminated. In Example 5, terms A and C are minor terms, while B is a



```
SEARCH: IFS EXAMPLE.

     IDENTIFICATION SECTION:
       SEARCHER = A N ANALYST.
       INQUIRER = A N CHEMIST: EXPERIMENTAL STATION: 1234.
       SEARCH_TITLE = DEMONSTRATION SEARCH.
     END_IS.

     LOGIC SECTION:
       SUBANSWER = FILMS + FIBERS.
       $FINALANSWER1 = SUBANSWER • 2002498(1).
       $FINALANSWER2 = (SUBANSWER + FINISHES) * DELUSTERANTS.
     END_LS.

     OUTPUT SECTION:
       ANS_FORM = (ABSTRACTS; SEARCH_TERMS).
       ANS_MAX = 250.
       ORDER(FINALANSWER2) = FILMS: FIBERS.
     END_OS.

END_SEARCH.
```

Note: 200249B is the Compound Registry Number for poly(ethylene terephthalate).

Figure 5. Sample information flow system search

Example 1:	Logic:	A	*	B		
	Number of Postings:	25		50		
Example 2:	Logic:	A	*	B		
	Number of Postings:	5000		50		
Example 3:	Logic:	A	*	B		
	Number of Postings:	5000		4000		
Example 4:	Logic:	A	•	(B	+	C)
	Number of Postings:	5000		50		80
Example 5:	Logic:	A	•	(B	+	C)
	Number of Postings:	25		4000		20

Figure 6. Different types of Boolean statements

major term. The expression will be changed into a union of A*B and A*C. The first expression will be evaluated by threading through the list (for A checking for B, while the second will be evaluated by threading through the list for) C, checking for A, as C has fewer postings than A. Again, duplicates will be eliminated.

The optimization techniques described above are extended to expand complex expressions into a series of logical tests, each of which contains a "most restrictive term." Specific account is taken of a number of special cases for which economies can be achieved. For example, if as the result of evaluation of several subanswers an expression must be evaluated which consists entirely of inverted lists or lists of QPD's which resulted from evaluation of the subanswers, instructions which will result in the simultaneous evaluation of up to 20 lists will be generated. The Search Edit program is by far the most complex of the search programs.

Several optional statements can appear in the Logic Section. These include a statement which requests that all links of a document be searched. Otherwise, except in subanswers, searching terminates as soon as one link is determined to be an answer. An optional statement limits the maximum amount of computer effort which can be expended on a single inquiry. Another type of statement will pass the results of a final answer to other searches in the batch, where they can be referenced in a manner analogous to the referencing of subanswers in final answers. A statement requesting that search output be saved is discussed later.

If the Output Section is omitted, search answers will be printed in only the form of a list of accession numbers. The most important statements specified in the Output Section concern answer format, output ordering, and a limit on the number of answers. The abstract format output request shown in Figure 5 will cause each answer accession number to be processed to produce abstracts printed in upper and lower cases. Another option causes each accession number to be written to a file for subsequent analysis or processing by special purpose programs. If the SEARCH_TERMS option is specified, terms in the final answers which cause each document to be an answer, will be printed following the accession number. When a union of several terms was specified, this option shows which term or terms caused the document to be an answer. Another format option permits the printing of accession numbers to be suppressed. All answer format options can be used together if desired.

The Order statement facility represents a significant increase in function over previous systems. Each final answer, or all final answers in the inquiry, can have their accession numbers printed in an order determined by the presence or absence of specified terms in the document. For example, the Order statement shown in Figure 5 will result in answers to the search being printed with documents which contain both the terms Films and Fibers first, followed by those with Films but not Fibers, followed by those with Fibers but not Films, and followed by all other documents. In each set, answers appear in ascending order of document number. The terms used in an Order statement need not be used elsewhere in the search. Order statements can be employed to apply weights to search answers. The results are weighted only in the sense that the appearance of one document before another in the answer owing to its containing more terms specified in the Order statement indicates greater relevance. If it is desired that documents containing any one of several terms appear together in an ordered answer output, these terms are enclosed in parentheses. If an Order statement were coded as A:(B:C), the first answer set would contain A and B or C; the second, A, but neither B nor C; the third, B or C, but not A; and the fourth, all other documents. Terms used in an Order statement can be chemical or general terms or subanswers. The ANS_MAX statement shown in Figure 5 will override the default limit to the number of answers which may be retrieved. The procedure followed if the limit is exceeded is discussed below.

Search Execution and Output Processing. In principle, the Search Execution program merely blindly executes the instructions prepared for it by the Search Edit program. Because the bulk of execution time is spent in the Search Execution program, it was written to a higher standard of performance than any other program. Separate modules handle threading of lists, merging of inverted lists, and generation of final output. In addition, during the execution phase, after a document is determined to be an answer to a search, the MAF or TMAF record is accessed to perform the ordering or search term specifications if present in the Output Section. These features are implemented by appending to the answer set document identification, consisting of QPD-link-internal accession number, two bit strings which represent each position within an Order statement, or a search term.

Control of search work space by Search Execution program is accomplished by maintaining a record of which locations in the Major Term File are available. These data are stored in the first five records of the Major Term File. During search execution, record locations are removed from the available space pool to store both subanswers and final answers. As each subanswer is processed for the last time, its record locations are returned to the pool, and may be used immediately again to store the final answer information. Answer records within the work space are chained together and do not necessarily occupy consecutive locations. During the output phase, the answer record locations are returned to the available space pool.

The Output Processing program generates a printed listing and the optional abstracts and file outputs. For each inquiry, the input text is reprinted, followed by any error messages. The bulk of processing in this phase consists of converting the internal accession numbers retrieved by the Query Execution phase to accession numbers as known to the user, sorting these by accession number or for the order option, and printing the results.

Generic Searching. The Central Report Index has a requirement for generic searching. A separate system is used for searching topological records of chemical compounds for substructures containing specified atom-bond relationships.[3] A file of chemical compounds described by fragment descriptors is also maintained. To understand how these searches interact with the IFS, it is only necessary to envision searches performed to find document number answers for the union of a large number of different terms, all of which have some attribute in common. For example, the terms may all be olefins or members of a specific dye class. Searches of this type are processed by a version of the Search Execution program which handles only unions of terms. The term codes are first processed by a program which looks up the head-of-list QPD for minor terms and the Major Term File references for major terms. The special search program processes the minor terms and major terms related to the concept separately. For minor terms, a threaded list search for up to 1000 terms is made simultaneously. Terms are arranged in ascending order of head-of-list QPD. Searching proceeds by examining each record specified on any of the threads. Even if several terms reference the same record, it is only necessary to retrieve this record once. Major terms are processed by extracting the inverted lists for each, and then merging these with the results of the threaded list search.

Output of the special search program is passed to the regular search routines, where the answers are referenced as pseudo-terms. For example, the convention used for substructure searches has the searcher code a term consisting of the letters RSS, which stand for Report Index Substructure Search, followed by the five-digit number employed to identify the substructure search during examination of the topological file. These pseudo-terms can be referenced in more than one IFS search, and can be saved if desired for reference in subsequent batches. Major Term File work space locations used to store the generic search answers are reused when the last reference to a generic search is processed during execution. A similar facility is used for searches of generic relationships among general terms, as found in the Thesaurus.

Special Search Features. The Information Flow System "Save" feature is able to retain results of searches for later use. A name and associated answer set address is held in a special file. Work space in the Major Term File used to store the answers is not returned to the available space pool. In the same or subsequent search batches, saved answer sets can be referenced as pseudo-terms. Saving of search answers may be initiated by a user or invoked automatically by the system.

A searcher may wish to save partial searches whose logic might be common to many searches. For example, many questions refer to families of fibers, such as Dacron polyester fiber, to analytical methods, and the like. Alternatively, different classes of terms might be combined, and then referred to in more than one search. For example, the users might desire to refer repetitively to documents containing Dacron terms relating to staple as opposed to dyeing of Dacron. A statement requesting that such searches be saved is coded in the Logic Section. The name to be associated with the answers is supplied by the searcher. If a search exceeds answer volume or cost limits, the system supplies a unique name. Such a saved answer set could be combined with more restrictive terms to reduce the number of answers.

The computer center in which the IFS is run bills for computer services on the basis of central processing unit time, memory size, input/output operations, and use of disk and tape drives. The cost is printed at the end of the job listing and is also charged to the account of the person submitting the run. Examining the magnitude of the costs is very helpful in optimizing a job before placing it in a production status.

To give some of this type of cost feedback information to persons submitting inquiries to the Information Flow System, a subroutine was developed which measures the use of computer system resources in the same manner as the computer center routine. At appropriate points in all search programs, when processing for an individual question begins or ends, a call is made to the routine which updates a field in memory. The true computer cost necessary to answer a question is printed at the bottom of the last page of answers. At the end of the run, a summary of these data is printed, along with overhead costs not associated with a particular question.

Searches are submitted from a teletypewriter terminal. Prior to the nightly search run, a run is made which checks the syntax of all submitted queries. The results of this check are transmitted back to the teletypewriter, after which corrections can be entered. If an error is discovered in a search, a message indicating the cause and the erroneous statement is printed. To avoid computer cost when the results might be incomplete, even one error will cause termination of that search. In addition, if the search passes an answer set to another search as discussed above, the related search is not executed either.

With such a large and complicated system, particularly one which would operate on a large data base, it seemed advisable to provide a mechanism for bypassing errors. The design of the IFS programs included provision for extensive error tracing and recovery. Almost all programs contain statements which are conditionally executed to produce information concerning internal processing. These statements may be activated without recompiling the programs, and provide an excellent means of analyzing program or file errors. In addition, the system is programmed to recover from anticipatable errors, such as input/output errors, inconsistent data, and the like. As a result, the search programs nearly always run to completion, even if errors occur. The occurrence of an error results in automatic printing of internal data which can be used to diagnose the problem.

Updating Logic. Updating of direct access files is completely different in both concept and implementation from the updating of sequential files. While many textbooks discuss problems encountered with sequential files, none discuss the class of problems encountered with the Information Flow System. File updating consists of two similar but basically separate processes: addition of new documents and modifications to existing documents. A flow diagram of the updating steps is shown in Figure 7.

New Document Addition. Addition of new documents consists of three steps. First, the new records are assigned to storage locations in the Temporary Master Accessions File. Second, information necessary to relate terms in new documents to the end of the threads for existing documents is generated. Finally, the new document records are formatted and added to the files. Initial input is submitted to the Document Translation program. This program retrieves internal accession numbers for the new document records from the Document Accessions File

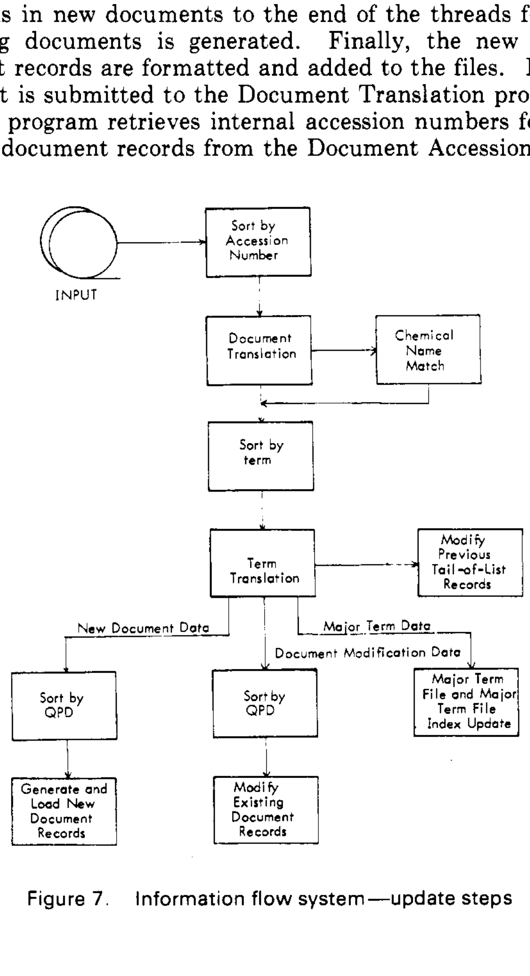

INPUT
Sort by Accession Number
Document Translation
Chemical Name Match
Sort by term
Term Translation
Modify Previous Tail-of-List Records
New Document Data
Major Term Data
Document Modification Data
Sort by QPD
Sort by QPD
Major Term File and Major Term File Index Update
Generate and Load New Document Records
Modify Existing Document Records

Figure 7. Information flow system—update steps

Index, and calculates the storage address for the indexing terms in the TMAF. The internal accession numbers are previously established using an auxiliary system to which the accession numbers of all new documents as they are received are submitted. This guarantees that an incorrectly punched accession number will not generate a master file record. Calculation of the storage address is done by estimating the record size based on the number of terms, roles, and links in each new document. A formula is available to compute how much space on a disk track is occupied by each record, including the overhead associated with each record. The program packs the TMAF tracks as completely as possible, and adds the track address and record key, combined as the QPD defined earlier, to each output record.

Indexing consists of general and chemical terms as discussed in the foregoing. While general terms are always indexed by name, chemical terms can be input either with a six-digit compound registry number or a name containing up to 48 characters. Names have been generated for this purpose for most of the chemicals which appear with significant frequency in the collection. Output records from the Document Translation program which contains chemical names are sorted by name, and matched against the file which contains a number corresponding to each name in the Chemical Name Match program. Output records from this program are recombined with general term and compound number indexing records.

The Term Translation program verifies that general terms submitted in the input have been defined as valid terms in the system by locating each general term in the Hierarchical File Index and extracting the internal term code for each valid entry. Compounds terms are checked to insure that they fall in one of the valid ranges defined for the system, and that the correct check digit was supplied. Separate output files for major and minor terms are generated for processing by subsequent phases. For minor terms, records for all documents which refer to the same term are chained together to produce a small threaded list consisting only of the documents which appeared in the update input. Because these documents will be appended to the end of the existing master file threads, the current tail-of-list field is accessed from the HF or CRF as appropriate, and replaced with the QPD of the last document appearing in the input. The list length field is also updated. It is then necessary to modify the document record referenced in the old tail-of-list field. A record is written out containing the term code, the QPD of the old tail-of-list record, and the QPD of the first new document record. This process is repeated for all input records referencing minor terms. In the next system step, the document which constituted the old tail-of-list is accessed. The field which would ordinarily contain the QPD of the next document on the thread is changed from an end-of-list indication to the QPD of the first new document. This record is then rewritten to the master file in its modified form. For major terms, a record containing the term code, role, QPD, and link is written out for input to the Major Term File Update program. Minor term records are written out containing the QPD, link, term code, role, and next and previous QPD's. The backward pointer is present because all new master file records are added to the TMAF. Records for the minor and major terms are sorted by QPD to collect all terms associated with a document in one place. The TMAF record is then generated and added to the file.

The Major Term File Update program is performed sequentially. Output of the Term Translation program is sorted into the same term-role-QPD-link order as the MTF inverted lists. The update program reads the Major Term File Index and the Major Term File, calculates new record location and posting count values, and writes out new copies of both files to replace the existing copies.

Existing Document Modification. Input records which will cause modifications to existing documents go through basically the same set of programs, with one exception, but the processes performed by those programs are different. Modifications can result in addition of a term to or deletion of a term from a record, addition of a role to or deletion of a role from a record, or transfer of a posting from one term to another within a document. Input records for existing documents are identical in format to those for new documents. The Document Translation program differentiates on the basis of whether a QPD has already been assigned to the document. For purposes of illustration, a transaction to add a new general term posting to an existing master file record will be discussed. Steps necessary for other types of transactions are similar.

The Document Translation program retrieves the QPD for the document from the Document Accessions File. The Term Translation program verifies that the term exists in the vocabulary, and substitutes the internal term code. The Term Translation program is also used to define new valid general terms prior to their use in indexing. For major terms, a record containing the term code and role, QPD and link are written out for processing by the Major Term File Update as discussed earlier. Transactions for minor terms are handled in a manner similar to that for minor term input for new documents. The records are linked together to form a separate threaded list consisting of all minor term additions for each different minor term. Unlike new documents, this small threaded list is added at the logical head of the existing threaded list for the minor term. The current head-of-list QPD is accessed from the HF or CRF and replaced with the QPD of the first document on the small threaded list. When the records are written out, the next QPD field for the last term on the small thread is the QPD of the old head-of-list document. This process is repeated for all minor terms which are being added to existing documents.

Output of the Term Translation program is sorted by QPD and processed by the Update Maintenance program. This program accesses the MAF or TMAF record referenced in each input record, and performs the necessary changes. Records are processed in QPD order so that each master file record has to be accessed only once, as multiple input records may reference a single document. When all changes have been made in the record, it is replaced in the master file.

As mentioned earlier, updating of files such as those maintained by the IFS can be quite complex. Problems not discussed here include recovery from input/output errors or logical errors which cannot be detected until the transaction is nearly complete. This category includes deletion of nonexistent terms, addition of a term to a document when that term already exists in the document, and addition of sufficient terms to a document to exceed the storage allocation. A separate solution was found to each of these and other problems.

OPERATION AND PERFORMANCE

The Information Flow System was designed to make maximum use of third generation hardware and software concepts. Tradeoffs in the development of the system were in the direction of minimizing execution time at some expense in memory size and disk storage utilization. The system runs on IBM 360/65 and IBM 370/155 computers operated by Du Pont's Central Systems and Services Department. Both machines use the MVT option of Operating System/360.

During the early stages of development, a programming language evaluation was made. PL/I seemed to offer important advantages to an information retrieval system such as IFS. In particular, character string manipulation, dynamic memory management, input/output capabilities, and list processing facilities seemed particularly attractive, and in fact, have been extensively exercised. A small number of assembler language routines comprise less than 2% of the system. The system was written using the PL/I(F) Compiler, and is in the process of being switched over to the recently delivered PL/I Optimizing Compiler. The programs which compose the system use different amounts of memory, with the update programs averaging approximately 140K bytes of core and the search programs averaging 160K bytes of core in the (F) Compiler versions.

Files for the Central Report Index application in December 1971 contained indexing for approximately 60,000 reports with 2.2 million postings to 160,000 chemical and general terms. Five IBM 2316 disk packs are used to store the data base.

Comparative performance data for a system such as IFS are not meaningful because they are almost entirely a function of three variables: the content of the files, particularly as it affects the number of terms and the distribution of document postings to those terms; the programs used, particularly reflecting the abilities and cleverness of the programmers and the degree to which generality was sacrificed for efficiency; and the economics of computer operation, particularly hardware utilization, peculiarities of the billing system, and availability of efficient devices for file storage. In addition, IFS economics are also affected by use of optional facilities such as output ordering and by the variability in complexity of questions. For these reasons, exact cost data are not given. It can be safely said that over-all economics were a significant improvement over the predecessor system to justify conversion. It is anticipated that costs will be reduced even more in the future owing to the reduced cost of file storage and computing, the availability of a better compiler for the programs, and fine tuning of the system based on operating experience.

FUTURE DIRECTIONS

As indicated earlier, the Information Flow System was designed as a total environment for processing of document-related data. Searches of second level files containing structural information about chemical compounds or generic relationships among general terms are currently performed using systems developed prior to the inception of IFS. These systems[1 3] operate independently of the IFS. A new system for processing chemical structure information is currently under development. The system will employ components of the Chemical Abstracts Service Registry System and a Du Pont-developed Screen Generation and Substructure Search facility. The files will be maintained and searched by modified versions of the IFS programs for processing document-related data. This system will be described in a future paper. An integrated Thesaurus facility will be developed as the final component of the originally specified IFS package.

In addition to the report application, it is anticipated that the IFS will be of use to other organizations throughout the Du Pont Company with similar objectives. A definite intention to utilize IFS has been expressed by one Du Pont industrial department to store information concerning correspondence, memoranda, and reports. Programs originally developed to convert the earlier system files to the IFS format have been refined and slightly generalized to facilitate future conversions.

The file processing techniques used in the IFS have significant potential for efficient processing of other classes of data. During 1972, it is planned to demonstrate how threaded lists and inverted files can be applied to information retrieval applications outside the classic document handling area.

When the original IFS concepts were developed, it was anticipated that improvements in computer technology would occur more rapidly than has been the case. Currently, it is not feasible to operate a system such as IFS in an on-line environment within Du Pont because of the very high on-line file storage costs. The system was implemented in such a way that it could be easily adapted to running in an on-line environment, as questions are basically processed separately one after another. However, extensive changes would be required to provide true "conversational" or "browsing" capabilities.

SUMMARY

Following a typically difficult conversion and start-up period, the Information Flow System is operating smoothly and satisfactorily. The system as configured closely resembles the original conception. A complementary system for processing second level chemical structure information is being developed. The true value of a systems development effort cannot be measured for several years. It is believed that the principles embedded in the IFS file structure and programs will permit the system to remain a useful tool for many years. The attention paid to modularity during design and implementation will allow the system to adapt to changing conditions of use and environment.

ACKNOWLEDGMENT

Many people contributed ideas and hard work to the Information Flow System project. Special mention is due to the implementation team which included Judith Armstrong, H. Terry Deamond, Sharyn R. Grim, Melvin L. Hill, Joanne Irish, Paul H. Pinson, Jr., and Janice L. Romstad.

LITERATURE CITED

(1) Montague, Barbara A., and Schirmer, Robert F., "Du Pont Central Report Index: System Design, Operation, and Performance," *J. Chem. Doc.* 8, 33 (1968).

(2) Lefkovitz, David, "File Structures for On-Line Systems," Spartan Books, New York, 1969.

(3) Hoffman, Warren S., "An Integrated Chemical Structure Storage and Search System Operating at Du Pont," *J. Chem. Doc.* 8, 3 (1968).