EXPERIENTIA

Vol. 27 – Fasc. 7 Pag. 745–872 15. 7. 1971

Applications of a Generalized Data-Management Language in Medicine: Some Recent Experience at Stanford University

By HERBERT R. LUDWIG

Polya 215, Stanford Computation Center, Stanford University, Stanford (California 94305, USA)

1. *Introduction*

This paper is concerned with the medical applications of an interactive file-oriented language that allows the user to interface with a text-editor and with his own FORTRAN code. The language was developed by J. VALLEE and implemented on the IBM 360/67 computer of the Campus Facility at Stanford University.

The system described in this paper was written in FORTRAN modules making it compatible with virtually any computer system that utilizes FORTRAN, accepting a few minor changes. Those persons interested in obtaining more information on the development and implementation should write or contact the author.

The first section of the paper is a brief description of the language. The four basic modes of operation – CREATE, UPDATE, QUERY, and STATUS – are discussed. Also, a brief discussion on file structures which will be useful for the prospective user of file-oriented systems.

The second section concerns itself with the different applications that DIRAC has had at the Stanford Medical Center, and the impact the introduction of such a system has had upon the cost and time spent on projects of similar nature.

Conclusions and some recommendations for the future are discussed in the third section. The impact of such a system upon the medical community, the patient and the doctor, with respect to its ability to accelerate the research process as well as treatment and diagnostic procedures are also discussed. Future recommendations are given in light of the experience gained from these studies.

2. *The DIRAC system*

2.1 Background. DIRAC (Date, Integer, Real, Alphanumeric, and Coded) is an information retrieval language which provides the user the ability to operate under four modes: CREATE, UPDATE, QUERY, and STATUS[1].

1. The CREATE mode allows the user to completely define the terminology and structure of his own file. The structure of the empty file is constructed under this mode.

2. The UPDATE mode allows such operations as adding, deleting modifying, or replacing records.

3. The QUERY mode of DIRAC allows the user to obtain information about selected subsets of his file at any level of the record structure so designated by the user during the CREATE mode. The different commands through which a file may be queried are described in a later section of this article.

4. The STATUS mode provides the user with an up-to-date status report for his particular file. Field identification, description of the fields, statistics and validation information are displayed in a standard report form.

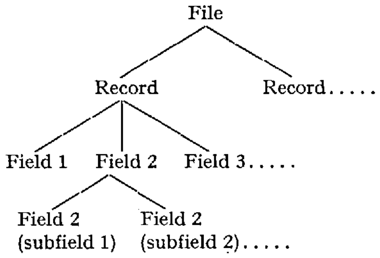
2.2 File structures for DIRAC. The medical field lends itself nicely to strictly defined file structures in that complex data-structures are usually not required, e.g., file structures beyond the subfield structure are usually not required. This has been the case at Stanford Hospital for the three applications discussed in this paper.

2.2.1 Fields and subfields. Within a DIRAC record every property is identified as an individual field: a patient's name in a hospital record, an address, an invoice number. Once record structure is determined by the user, the fields are declared to DIRAC and named and identified during file creation. They are then available for any type of retrieval response from the file. Fields of a record can be numeric interger such as 'Age', numeric real such as 'purchases' on a charge account (XX.XX), alphabetic such as 'Name' or 'Address'; they can also be dates or codes. (Codes are to be implemented in a later version of DIRAC.)

A record consists of fields which may themselves be formed from two or more subfields. This process of

[1] J. VALLEE and H. R. LUDWIG, The DIRAC *Language: Concepts and Facilities*; Research Report No. 1. Information Systems Group, Stanford University Computation Center (May 1970). – J. VALLEE and H. R. LUDWIG, DIRAC: *An Overview of an Interactive Retrieval Language*, Educ. Newsletter (September 1970).

subdivision (tree structure) can theoretically be continued.



However, in the first version of DIRAC file structure representations were not supported beyond the subfield level.

2.2.2 File construction under DIRAC. DIRAC provides the user with the opportunity to completely specify his own file organization or structure. Thus, the user does not have to be concerned about using a fixed field or fixed word length of format. The user is not bound by a set of rigid rules pertaining to record size, length, etc., and these parameters are not even apparent to him.

The user should first compile a working list of all fields to be included in a record, specifying whether or not a field is singular or multiple (subfields). Example: Suppose that we want to create a DIRAC file of heart-transplant patients; we have determined that we want to include the following information (fields) in a patient's record:

Field description	Single/ Multiple
Name of patient	S
Hospital record number	S
Home address	S
Date of birth	S
Home phone	S
Cardiac diagnosis	M
Vascular diagnosis	M
Admission dates	M
Dates of operations	M
Status at time of discharge	M
Date of death	S

A typical patient record would have the structure:

Name	Record #	Address	Birthdate	Home phone
JOHN L. SMITH	237863	33 Fifth Ave. Glendale, Ca.	19080325	459–7414

Cardiac diagnosis	Vascular diagnosis	Admission dates	Operation dates	Status	Date of death
(text)	(text)	19680325	–	(text)	
(text)	(text)	19681103	19681120	(text)	
.	.	.	.	.	
.	.	.	.	.	

Note that the fields Cardiac diagnosis, Vascular diagnosis, Admission dates, Operation dates, and Status at time of discharge are multiple. In other words, a patient might have been admitted several times for some type of diagnostic checkup or operation over the past year(s); each time the patient was admitted pertinent data was recorded in the appropriate field.

The user must also determine the 'type' of each field which he includes as part of a record. For example, patient's name would be alphanumeric (ALPHA), whereas hospital record number would be INTEGER; date of birth would be DATE type, as well as admission dates and dates of operations. However, in the latter two cases a distinction is made to depict the MULTIPLE status of these two fields.

After determining the type of each field and whether or not that field is singular or multiple, the fields may be numbered and given a name as follows:

Field	Name	Description
1	Name	Name of patient
2	Number	Hospital record number
3	Address	Home address
4	BirthDate	Date of birth
5	Phone	Home phone
6	CDiagnosis	Cardiac diagnosis
7	VDiagnosis	Vascular diagnosis
8	AdmDate	Admission dates
9	OpDate	Operation dates
10	Status	Status at time of discharge
11	DDate	Date of death

A delimeter will be picked from a set of special characters (such as #, $, ||) to denote a field in DIRAC. (The user can pick any delimeter out of the list which is convenient to him, thus avoiding the need for a rigid standard notation imposed by most existing systems. If the user does not select a notation for record and field, the standard notation $ and # are used respectively.)

DIRAC will prompt the user for type and multiplicity of the fields within a record. In our example the following information would then be typed at the terminal (where every line beginning with a colon is typed by the user):

```
7.  SUPPLY DATA TYPE AND MULTIPLICITY
    :      ALPHA SINGLE #3 #1 #5
    :      INTEGER SINGLE #2
    :      DATE SINGLE #4 #11
    :      DATE MULTIPLE #8 #9
    :      ALPHA MULTIPLE #6 #10 #7
```

The user should note that field specifications can be input in any order. Also note that the delimeter '#' was used to specify fields. 'Integer Single' means that the value to be stored in field 2 will be a single integer number. 'Alpha Multiple' means that there exists a multiple field in which alphanumeric information is stored. (The same holds true for a DATE MULTIPLE specification.) From the example we note that fields #6–#10 are multiple. Thus, when reference is made to #1 – admission date – the cardiac diagnosis, vascular diagnosis, date of operation (if there was an operation) for that visit (or duration of stay), and a statement concerning the condition of the patient at time of discharge are contained in #6[1], #7[1], #9[1], and #10[1], respectively.

2.2.3 Medical files. Most files encountered in the medical field have the following properties in common: 1. They are patient oriented files, i.e., they are set up by patient name. 2. Each patient has fields which are similar to other patients, even though these may be in a completely different file, e.g., address, phone, hospital I.D. number, blood type, etc. 3. Most files do not go beyond the subfield structure, e.g., 'treatment' might be the name of a field which is designated ALPHA MULTIPLE. This means that alphanumeric information is stored in this field and that successive information within each subfield (such as in the case of an array item in FORTRAN) refers to a treatment given the patient on a given date, whereas 'RxDate' might be the name of a multiple field which stores the dates in each successive subfield for each successive treatment respectively.

3. *DIRAC applications in the hospital environment*

3.1 Hematology clinic. 3.1.1 Statement of problem: The rapidly expanding volume of bone marrow examinations performed at Stanford Medical Center has presented many problems for the physician with respect to his daily work patterns. These problems, for which no classical solution is available, have only recently been recognized. The ability to store and retrieve pertinent patient record information is a case in point. Since it is essential to be able to analyze these data in order to perfect new methods of diagnosis and treatment of hematologic disease, the use of a file-oriented language seemed inevitable. By the introduction of DIRAC the following objectives were met: 1. The production of a simple bone marrow report format showing all of the essential information necessary for meaningful identification and analysis. 2. Interfacing this report generator with an efficient file organization within the DIRAC environment.

A brief description of the problem of data-base implementation under these constraints is now presented along with an example of 'conversational' interrogation.

3.1.2 Data-base implementation: A description of the bone marrow report system is presented in Figure 1. Note that the bone marrow reports were typed by a secretary and then filed manually either by a clerk or by the secretary. This method proved to be quite cumbersome because when specific bone marrow reports were desired for study a manual sort had to be made to retrieve the document(s). This was often time consuming and a very burdensome task. Note also that there existed no feedback to the patient from the bone marrow reports after they were filed.

Figure 2 describes the first phase in automating the bone marrow report system described in Figure 1. The bone marrow diagnosis was typed into a working data set in computer storage. The reports could now be checked for accuracy and verified. Additions and corrections could be made with ease utilizing WYLBUR[2], the Stanford text editor. A report generator was written which would generate a formated general report for each patient record. Copies of these reports

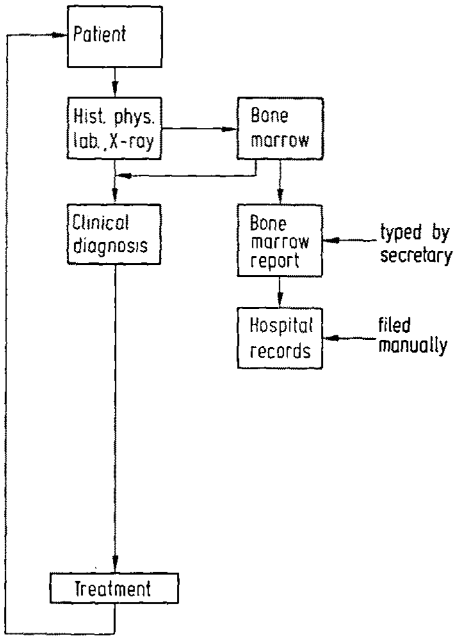


Fig. 1.

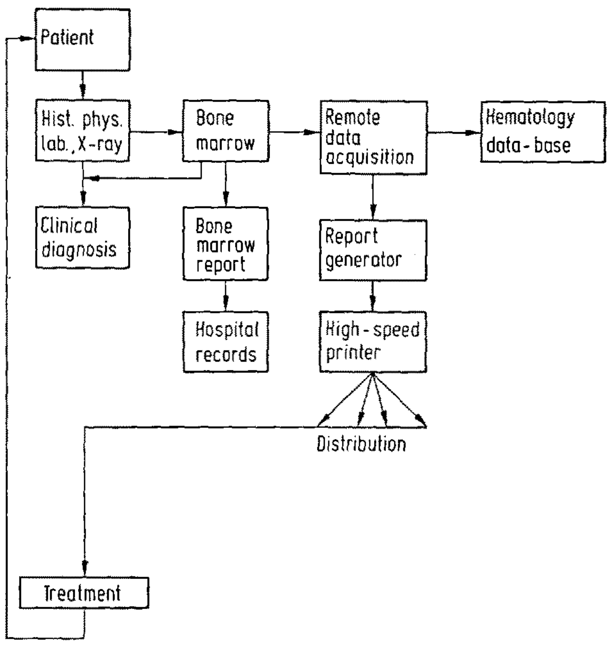


Fig. 2.

[2] R. FREDRICKSON, WYLBUR on the IBM 360/67; Stanford University Computation Center, Stanford, California (1969).

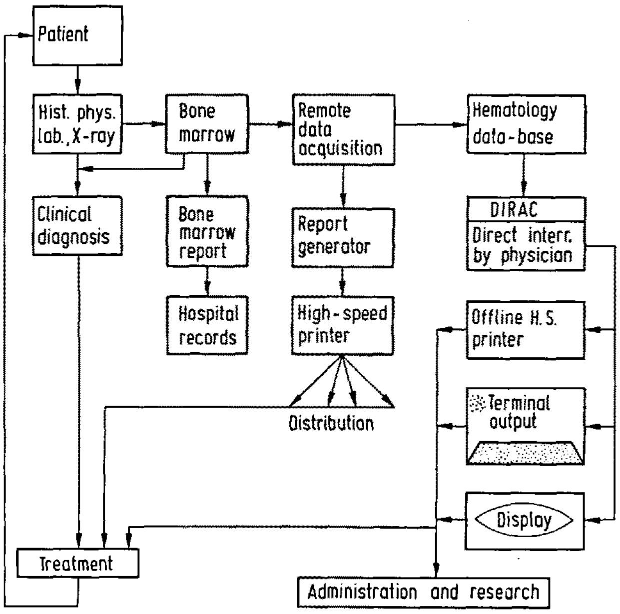


Fig. 3.

are now distributed to different physicians and to the medical records section of the hospital.

This automatic report generating system was a vast improvement over the simplified file system described by Figure 1. The problem of distributing bone marrow reports to various departments was now solved. Since the bone marrow data was now being input remotely into the computer memory, there now existed a database within the computer. Thus, the problem of storage and retrieval of specific patient information when needed from the bone marrow file had been partially solved – a data base existed in machine readable form – but there existed no facility to interrogate the file. If a physician wished to have information regarding a specific patient or obtain statistical data concerning certain parameters within the records he still had to search the file manually. The clerical task of retrieving records for review and research analysis was solved by the introduction of DIRAC, thus, transforming Figure 2 into Figure 3. Note that there now is feedback to medical personnel on the diagnosis. This feedback

Stanford University Computation Center — Status report for file M010 (Class 1), 1 July 1970 — Language: DIRAC-1A

Description: Bone marrow report

Creation date: 15 May 1970	1 = Type	File name:	M010
Record notation: N	2 = Multiplicity	File created by:	Wolf
Field notation: #F	3 = Indexing	Record length:	512
Disposition: Private	4 = Code residence	No. of fields:	29
No. of records: 235	5 = Code type	Latest update on:	15 June 1970

Field identification			Statistics and validation information													
	Field name	Description	Storage					Validations			Statistics			Existence	
			1	2	3	4	5	Nec.	Size	Sub.	Size	Dec.	Sub.	PFC	PCT (%)
1	Name	Name of patient	A	S	0			yes	0	0	24	0	0	235	100.00
2	Sex		A	S	0				0	0	6	0	0	235	100.00
3	Birth	Birthdate	A	S	0				0	0	9	0	0	234	99.57
4	Ward	Ward No.	A	S	0				0	0	10	0	0	235	100.00
5	Phys	Ref. physician	A	S	0				0	0	13	0	0	218	92.77
6	Record	Medical record No.	A	S	0			yes	0	0	9	0	0	235	100.00
7	Marrow No.	Bone marrow No.	A	S	0			yes	0	0	8	0	0	235	100.00
8	Date	Examination date	A	S	0				0	0	8	0	0	235	100.00
9	Previous	Previous marrow	A	S	0				0	0	3	0	0	157	66.81
10	Clinical	Clinical information	A	S	0				0	0	144	0	0	219	93.19
11	RBC	Peripheral smear section	A	S	0				0	0	363	0	0	208	88.51
12	Platelets	Peripheral smear section	A	S	0				0	0	57	0	0	202	85.96
13	WBC	Peripheral smear section	A	S	0				0	0	139	0	0	200	85.11
14	C1	Peripheral smear section comment	A	S	0				0	0	67	0	0	43	18.30
15	Quality	Bone marrow aspirate section	A	S	0				0	0	31	0	0	230	97.87
16	Ratio	Myeloid/erythroid – Bone marrow Se	A	S	0				0	0	39	0	0	194	82.55
17	Meg	Megararyocytes – Bone marrow Se	A	S	0				0	0	73	0	0	217	92.34
18	Myeloid	Myeloid elements – B.M. section	A	S	0				0	0	144	0	0	217	92.34
19	Erythroid	Erythroid elements – B.M. section	A	S	0				0	0	96	0	0	217	92.34
20	C2	Bone marrow section comments	A	S	0				0	0	188	0	0	180	76.60
21	Amount	Iron stores section	A	S	0				0	0	32	0	0	195	82.98
22	Location	Iron stores section	A	S	0				0	0	37	0	0	116	49.36
23	C3	Iron stores section comments	A	S	0				0	0	56	0	0	39	16.60
24	Impression	General comments	A	S	0				0	0	268	0	0	228	97.02
25	Qual	Spicule section	A	S	0				0	0	159	0	0	82	34.89
26	Confirms	Confirms other data – Spicule section	A	S	0				0	0	3	0	0	115	48.94
27	C4	Spicule section comments	A	S	0				0	0	222	0	0	130	55.32
28	Hematologist		A	S	0			yes	0	0	19	0	0	235	100.00
29	Director		A	S	0			yes	0	0	25	0	0	235	100.00

Fig. 4

is practically instantaneous, and thus, alleviates the clerical burden of searching manually for records within the file.

3.1.3 Conversational interrogation. Another advantage gained by introducing a data-management system is that it allows research and administrative procedures to be simplified. A whole file of bone marrow reports is now available for interrogation by medical researchers. By use of its interface capability DIRAC can perform statistical analyses on pertinent records of the file when directed to do so. Interrogation by users of the file can also be displayed on a video unit or typed out at the user terminal, or can be printed on a high speed printer, whichever is more convenient based on the query and need of the user. Figure 4 is a status report for the Bone Marrow File. Note field identifications, statistics, and validations given to the user.

The following example shows how the user can carry on a simple 'conversation' with DIRAC. Browsing the file, retaining subsets of selected queries, and displaying pertinent information are all displayed. (Each line beginning with a colon is the user's response to a DIRAC prompt).

In the above example the exclamation mark stands for a 'wild' character. Thus, searches may be made over a specific set of digits in this case. Also the use of logical expressions is valid.

3.2 *Stanford Blood Bank*

3.2.1 Statement of problem. The Stanford Blood Bank, headed by Dr. PAUL WOLF, has initiated a research effort leading to a complete model of an automated transfusion center. This research effort leads to results that will be generally applicable throughout the clinical laboratory.

Even with the advent of better transfusion techniques and devices, the Hospital still envisions an ever increasing use in whole blood. Increased patient traffic, due to the advent of new technology and greater capacity of the hospital, will add to this expansion. Thus, a control system that will govern the use and allocation of blood units throughout the hospital is a definite need. Along with this increase in whole blood use will come an increase in the paper load both for administrative and research purposes.

With the introduction of DIRAC, the above problems can be directly confronted and solved. The 'non-procedural' technique is characterized here by the ability for the blood bank clerk, the research assistant, medical technologist, or a doctor, etc., to directly interrogate or access a specified file without programmer intervention or programming knowledge.

3.2.2 Systems design and analysis. The objectives of the project were threefold: 1. To decrease the paper-volume and paper-traffic through the blood bank. 2. To design an information system which will allow medical personnel to access the files directly[3]. 3. To set up inventory and control systems for all units passing through the blood bank[4,5].

The system as it existed prior to the design phase is described in Figure 6. Note that many of the services performed by the clerks could easily be performed by the computer, e.g., updating donor cards, the entering of recipient information in the S.U.H. logbook, etc. The main files in this system are the Stanford Donor File, the Recipient File, and the Two Logbooks – Stanford University Hospital Logbook and the Blood-from-Other-Banks Logbook.

Figure 7 shows how the existing system was redesigned. Note the three sequences into which the system has been partitioned: the matching, inventory, and update sequence. The flow of paper is now administered internally by the blood bank clerk through the use of this data-base management system.

```
:   SELECT ALL
    ---------------------------
    401 RECORDS SELECTED
:   Sex CONTAINS FEMALE END
    ---------------------------
    168 RECORDS SELECTED
:   RETAIN
:   MarrowNo CONTAINS 77 END
    ---------------------------
    27 RECORDS SELECTED
:   Director CONTAINS WOLF END
    ---------------------------
    3 RECORDS SELECTED
:   TYPE Name MarrowNo END
    ---------------------------
    55
Name        MARY Q. SMITH
MarrowNo    55-77-25
    ---------------------------
    96
Name        JOAN R. JONES
MarrowNo    77-22-28
    ---------------------------
:   RELEASE
:   Impression CONTAINS 'HODGKINS DISEASE' END
    ---------------------------
    50 RECORDS SELECTED
:   Date CONTAINS '3-1!-70' AND
    Quality CONTAINS GOOD END
    ---------------------------
    2 RECORDS SELECTED
:   TYPE Name Record Date Quality Impression END
    ---------------------------
    106
Name        JOHN R. HOPKINS
Record      453-789
Date        3-16-70
Quality     GOOD
Impression  EOSINOPHILIA. PLASMACYTOSIS AND
            INCREASED R.E. CELL IRON
    ---------------------------
    368
Name        KENNETH M. LAWRENCE
Record      583-91
Date        3-11-70
Quality     GOOD
Impression  NON-DIAGNOSTIC MARROW. PLASMA CYTOSIS
            AND MILD EOSINOPHILIA.
    ---------------------------
:   RELEASE
        .
        .
```

Fig. 5. Example of conversational query.

[3] J. M. STENGLE, Transfusion *10*, 3 (1970).
[4] C. F. HOGMAN and O. RAMGREN, Transfusion *10*, 3 (1970).
[5] R. A. STEWART and W. B. STEWART, Transfusion *9*, 2 (1969).

3.2.3 Emergency interrogation of Blood Donor File. Assume that an emergency has arisen at a participating hospital. The Stanford blood bank receives a call to supply or locate donors with 0 NEG blood type. The problem for the blood bank is to check their inventory and determine whether or not they can afford to delete their supply of 0 NEG blood or locate donors and send them as fast as possible to the participating hospital. Assume that the blood bank staff decides on the later alternative. A clerk will immediately establish communication with the computer (at night, a terminal located in the director's home can be used for this purpose) and the interrogation of the donor file takes place as follows (where every line beginning with a colon is typed by the user):

The field 'Serum' is tested to determine whether or not the blood-type is 0. The resulting 512 records are retained for further interrogation. 'RHType' is now tested and 28 records are found to exist with an RH factor NEG. A successive filtering on each set of resulting records obtained from each query results in

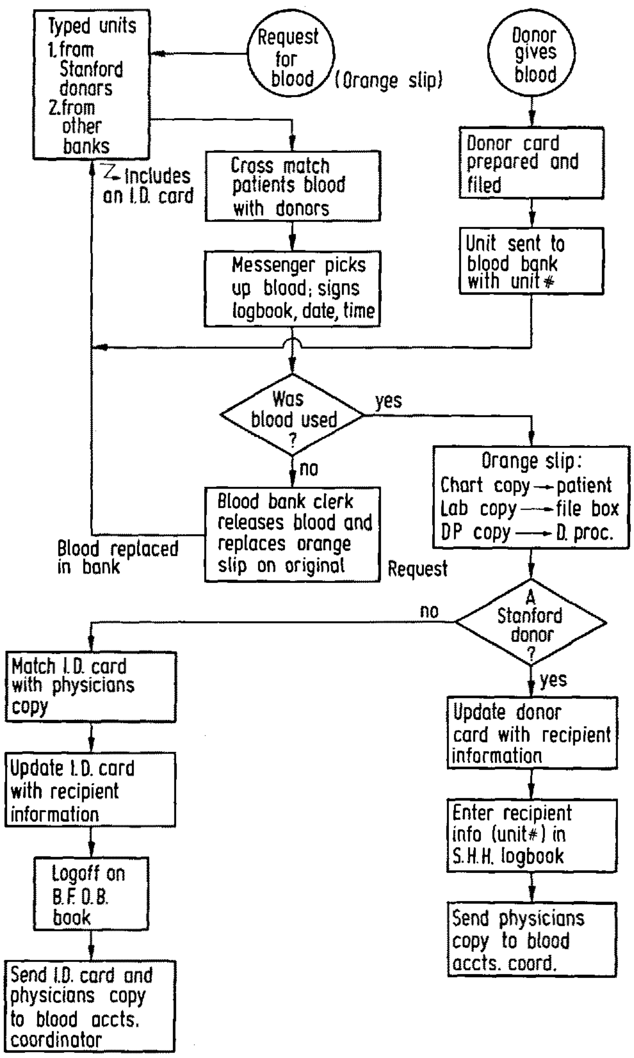


Fig. 6.

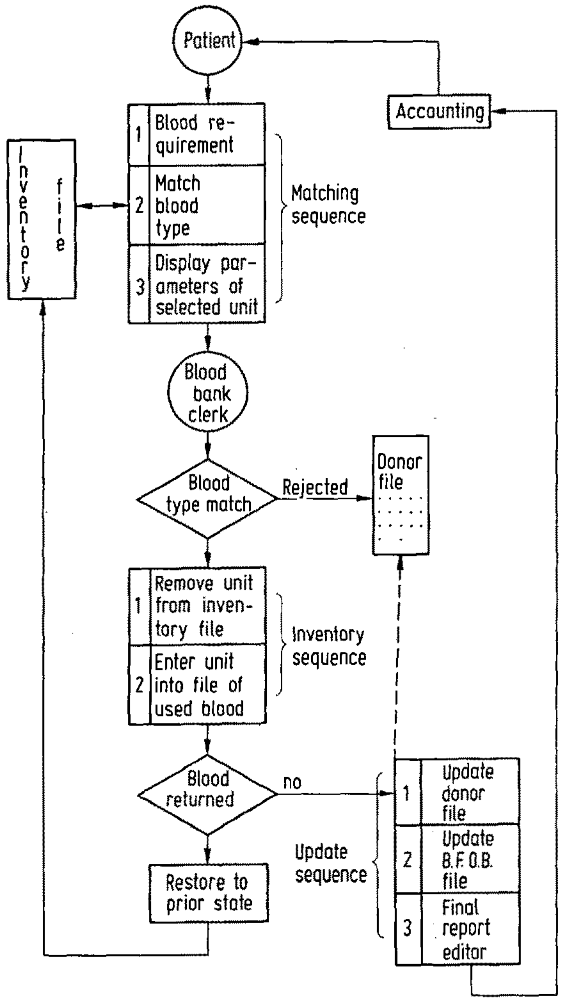


Fig. 7.

: Serum CONTAINS 0 END

512 RECORDS SELECTED
: RETAIN
: RHType CONTAINS NEG END

28 RECORDS SELECTED
: City CONTAINS 'Belmont' END

20 RECORDS SELECTED
: DIFFERENCE (PresentDate – LastDate) > 60 END

18 RECORDS SELECTED
: HomePH CONTAINS '593–' END

11 RECORDS SELECTED
: TYPE Name Address City HomePH Bus.PH Serum RHType END

23
Name BEST, JOHN S.
Address 1721 Hiawatha Dr.
City Belmont
HomePH 593–1789
Bus.PH 488–9161
Serum 0
RHType NEG

55
Name SMITH, ROGER D.
Address 275 Fifth Ave.
. .
. .
. .

Fig. 8. Example of emergency interrogation by blood bank.

the selection of 11 records. The purpose of this query was to find all those persons whose blood type was 0 NEG, who lived in Belmont, who hadn't given blood in the last 60 days, and whose home phone started with digits '593–'. The blood bank clerk can now print out the pertinent parameters for these individuals (phone no.), call them, and send them to the participating hospital to donate blood. This problem could be solved even further if the participating hospital had at its disposal a terminal which communicated with the

Date

STANFORD UNIVERSITY HOSPITAL
BLOOD BANK
STANFORD, CALIFORNIA 94305

Donor Number

Name ______ (LAST) ______ (FIRST) ______ (MIDDLE) ______ Birth Date ______ Minor? ______ Sex ______

Address ______ City ______ Home Ph. ______

Date last donation ______ No. of donations in last 12 mos. ______ Bus. Ph. ______

Credit to ______

CBR ______

Successful bleeding ______ Unsuccessful ______ Why ______

Reaction ______

REMARKS: ______

M.D. ______ Phlebotomist: ______

Hemantigen: ______

Serum Grouping ______ Cell Grouping ______ RH Typing ______ Serology ______

Miscellaneous: ______

Technologist: ______

Expiration Date ______ Recipient ______ Hospital ______

HISTORY Donor: Do you have or have you ever been subject to the following diseases and conditions?

	NO	YES		NO	YES		NO	YES	PHYSICAL EXAM
Malaria			Rheumatic Fever			Dental Ext. (Last 72 Hrs)			TEMP.
Malaria Area or Drugs			Kidney Disease			Under Physician Care			PULSE:
Jaundice, Liver Disease, Hepatitis			Infectious Mononucleosis			Medications			B.P.:
Jaundice or Hepatitis Contact			Swelling of Feet			Immunization or Injection *			WEIGHT:
Allergies			Convulsions, Fainting Spells			Surgery			ANT. CUB. FOSSA
Skin Eczema, Boils, Dermatitis			Unexplained weight loss			Hospitalization			HGB:
Ulcers			Shortness of Breath			Other illness within 1 Yr.			$CuSO_4$
Tuberculosis			Pain in Chest			Pregnancy			CYANMETH
Diabetes			Heart Condition			Transfusions			
Hazardous Occupation			Recent Cold or Sore Throat			Venereal Diseases			
Undulant or Prolonged Fevers			Persistent Cough			Food or Drink within 4 hrs.			
Bleeding Tendency			Any Malignancy			Tattoos			

* INCLUDES IMMUNIZATION TO HUMAN BLOOD CELL ANTIGENS

REMARKS:

STATEMENT AND RELEASE

I hereby certify that I have been questioned concerning the diseases, illnesses, symptoms and other physical conditions appearing on this card; that I have answered truthfully concerning each of the same and all questions asked me and that I have never suffered any of said diseases, illnesses or symptoms, except as indicated

I voluntarily donate my blood to the STANFORD UNIVERSITY HOSPITAL BLOOD BANK, STANFORD, CALIFORNIA 94035, to be used as decided by the said Blood Bank I hereby release the Stanford University Hospital, its members, agents, representatives and employees from all claims and demands whatsoever which I have or may have by reason of the taking of blood to which I have submitted or am about to submit, and consequences resulting therefrom.

______
Donor's Signature

Accepted ______

Refused ______ Why ______ M.D. ______

Fig. 9.

Stanford University Computation Center | Status report for file M200 | (Class 1) | Language: DIRAC-1A
28 September 1970
Description: Blood donors at Stanford hospital

Creation date: 18 September 1970
Record notation: \$N\$
Field notation: #F
Disposition: Private
No. of records: 0

1 = Type
2 = Multiplicity
3 = Indexing
4 = Code residence
5 = Code type

File name: M200
File created by: Wolf
Record length: 512
No. of fields: 33
Latest update on: 18 September 1970

Field identification			Statistics and validation information												
Field name		Description	Storage					Validations			Statistics			Existence	
			1	2	3	4	5	Nec.	Size	Sub.	Size	Dec.	Sub.	Rec.	PCT (%)
1	Name	Name of donor	A	S	0			yes	0	0	0	0	0	0	0.0
2	Birthdate	Date of birth	D	S	0				0	0	11	0	0	0	0.0
3	Minor	Under 21 years of age	A	S	0				0	0	0	0	0	0	0.0
4	Sex		A	S	0				0	0	0	0	0	0	0.0
5	Address	Address of donor	A	S	0			yes	0	0	0	0	0	0	0.0
6	City		A	S	0			yes	0	0	0	0	0	0	0.0
7	HomePH		A	S	0			yes	0	0	0	0	0	0	0.0
8	BusinessPH	Business phone	A	S	0				0	0	0	0	0	0	0.0
9	Date	Date of donation	D	S	0			yes	0	0	11	0	0	0	0.0
10	Number	Blood bank number	I	S	0				0	0	0	0	0	0	0.0
11	Last date	Date of last donation	D	S	0				0	0	11	0	0	0	0.0
12	Donations	No. of donations in last 12 mos.	I	S	0				0	0	0	0	0	0	0.0
13	Credit	Credit to...	A	S	0				0	0	0	0	0	0	0.0
14	CBR		A	S	0				0	0	0	0	0	0	0.0
15	Bleed	Succ. and unsucc. and remarks	A	M	0				0	0	0	0	0	0	0.0
16	Reaction	Type of reaction	A	S	0				0	0	0	0	0	0	0.0
17	Remarks	General comments	A	S	0				0	0	0	0	0	0	0.0
18	M.D.	Doctor present	A	S	0				0	0	0	0	0	0	0.0
19	Phlebotomist	Phlebotomist present	A	S	0				0	0	0	0	0	0	0.0
20	Hemantigen		A	S	0				0	0	0	0	0	0	0.0
21	Serum	Serum grouping	A	S	0				0	0	0	0	0	0	0.0
22	Cell	Cell grouping	A	S	0				0	0	0	0	0	0	0.0
23	RhType	Rh typing	A	S	0				0	0	0	0	0	0	0.0
24	Serology	Serology	A	S	0				0	0	0	0	0	0	0.0
25	Misc	Miscellaneous	A	S	0				0	0	0	0	0	0	0.0
26	ExpireDate	Expiration date	A	S	0				0	0	0	0	0	0	0.0
27	Rechnologist		A	S	0				0	0	0	0	0	0	0.0
28	Recipient		A	S	0				0	0	0	0	0	0	0.0
29	Hospital		A	S	0				0	0	0	0	0	0	0.0
30	History	History of donor	A	M	0				0	0	0	0	0	0	0.0
31	Physical	Physical examination	A	M	0				0	0	0	0	0	0	0.0
32	Immunization	Inc. imm. to human bld. cell ant	A	S	0				0	0	0	0	0	0	0.0
33	Comments	General remarks	A	S	0				0	0	0	0	0	0	0.0

Fig. 10

Stanford Blood Bank. Thus, the particpating hospital could directly interrogate the Stanford Blood Bank Donor Files and locate the proper individuals. This network concept of information retrieval has been under study and results are forthcoming[6].

Querying a file in this manner seems highly advantageous as opposed to searching by hand through a card file. The front and backside of the donor information card are shown in Figure 9. Attempting to search through a few thousand donor cards for specific blood types could become a tedious and highly burdensome task. Having the ability to interrogate this information internally stored in the computer gives the clerk a freedom that can be applied to more meaningful tasks.

3.2.4 Management reports. Figure 10 is a description of the donor file as it now exists in prototype form. This is a standard report form which is printed out by DIRAC at the user's discretion. Note that the report gives all vital statistics of the file such as creation date, disposition of the file, who created the file, and information on its contents, as well as internal records information necessary for updating and querying the file. The

```
NAME OF USER
  :   SMITH
PLEASE TYPE EXECUTION MODE
  :   UPDATE
FILE IDENTIFICATION
  :   X990
FIXED FORMAT ?
  :   NONE
INTERFACE ?
  :   WYLBUR  <-----
MAXIMUM RECORD LENGTH :  326 WORDS.
THE FILE CONTAINS    4067 RECORDS.
UPDATE EXECUTION TERMINATED.
          .
          .
```

WYLBUR data set containing original file records in working storage area. See Figure 12.

Fig. 11. Example of update mode in DIRAC.

[6] J. VALLEE, *Scientific Information Networks: A Case Study*; Research Report No. 2. Information Systems Group, Stanford University Computation Center (October 1970).

user also has the option of writing other formats of reports which would be meaningful to him in his work and then querying the file and selecting the necessary records and displaying them through this medium. In other words, the user is limited only by his own imagination as to what data he can display through the various types of DIRAC interfaces available to him (Terminal, Scope, and WYLBUR).

3.2.5 Interactive updating of files. The environment under which the user can update files or records within files is unique to DIRAC. The user has the option of either updating his file on a real time basis by terminal interaction or by working through an interface provided by the text editor, WYLBUR. Thus, if the user wishes he can catalogue his updates and drive them (update the file) through WYLBUR. The following example will demonstrate this procedure thoroughly: Each line beginning with a colon refers to the user's response (Figure 11).

The user identifies himself and specifies the execution mode which he will be using. Identification for the file is M010 (Bone Marrow) and the format of the file is not fixed. DIRAC now prompts the user to designate whether updating will take place through WYLBUR or terminal interface. At this point if the user's data set (where the records for updating should be – in WYLBUR) is empty, DIRAC will prompt the user and tell him that his data set is in fact empty and that he now is given the opportunity to bring those records into his data set either from the disk where they are stored or typing them now. (The latter alternative is, of course, an option, however, the reader should realize that this procedure would then be the same as terminal interaction on the part of the user.) The real advantage in using WYLBUR for updating procedures is that large amounts of data can be input into the file without the user having to sit at the terminal and type in each record. Of course, the record information has to be keypunched somewhere, but by placing it in a working storage area which WYLBUR provides, the user can edit his information prior to update making sure that most typing errors, etc. are corrected. Also, terminal time would be expensive, as well as CPU time if the user were to sit at the terminal and input the record information without going through the WYLBUR interface.

An example of the data and how it might appear in WYLBUR ready for input into DIRAC is shown in Figure 12.

The user may wish to delete, revise, or replace a record in his file. Usually these revisions or changes are of a lesser magnitude than those of initially inputing – updating – the data into the file, i.e. using the WYLBUR interface. However, there may be times when the user might desire to use WYLBUR for this type of update. In the following example the terminal interaction is displayed only. Each line beginning with a colon refers to the user's response (Figure 13).

In the above example the user first designates that he is going to update 'OLD' records. Thus, he is replacing record number 1 with another record or maybe a similar record with part of the field information altered. He then designates the mode 'NEW' which means that he is going to input a new record into the

```
NEW
#1  'MARKSON. JAMES'
#2  MALE
#3  '12-10-30'
#4  'E1A'
#5  HOLMAN
#6  '45-10-26'
#7  'A70-845'
#8  '7-24-70'
#9  YES
#10 'ACUTE MYELOGENOUS LEUKEMIA'
#14 'NONE SUBMITTED'
#15 'SCANTY SPECIMEN'
#17 'NONE SEEN'
#18 SPARSE
#19 SPARSE
#24 'MARROW INFILTRATED WITH MYELOMONOBLASTS;
    ? SLIGHT INCREASE IN MATURITY OF BLASTS SINCE
    LAST MARROW'
#27 'NONE SUBMITTED'
#28 'ABRAHAM POTOLSKY, M.D.'
#29 'GEORGE WALTUCH, M.D.'
NEW
#1  'SIMM, ROBERTA A.'
#2  FEMALE
#3  '4-21-36'
#4  'E1B'
#5  'J. WARTMAN'
#6  '34-60-87'
#7  'C70-765'
#8  '8-21-70'
#9  NO
     .
     .
     .
```

Fig. 12. Example of WYLBUR update file.

```
     .
     .
FIXED FORMAT ?
  :    NONE
NEW OR OLD ?
  :    OLD
ACTION
  :    REPLACE $1$
RECORD NO.     1 DEL.
  :    #1 JOHN R. SMITH
  :    #2 585 El Camino, Palo Alto, California
          .
          .
          .
  :    NEW
RECORD NO.     4068
  :    #1 LARRY S. JONES
  :    #2 45 Fifth Ave., New York
          .
          .
          .
  :    END
MAXIMUM RECORD LENGTH :  326 WORDS.
THE FILE CONTAINS 4068 RECORDS.
UPDATE EXECUTION TERMINATED.
     .
     .
```

Fig. 13. Interactive updating of DIRAC file.

Stanford University
Computation Center

Status report for file M300 (Class 1)
28 September 1970

Language: DIRAC-1A

Description: File of prostate cancer patients

Creation date: 16 September 1970
Record notation: SNS
Field notation: #F
Disposition: Public
No. of records: 139

1 = Type
2 = Multiplicity
3 = Indexing
4 = Code residence
5 = Code type

File name: Prostate
File created by: Ray
Record length: 1024
No. of fields: 68
Latest update on: 16 September 1970

Field identification			Statistics and validation information												
Field name		Description	Storage					Validations			Statistics			Existence	
			1	2	3	4	5	Nec.	Size	Sub.	Size	Dec.	Sub.	Rec.	PCT (%)
1	Name	Patient's name	A	S	0			yes	0	0	22	0	0	139	100.00
2	SUH#	Stanford university hosp. No.	A	S	0			yes	0	0	8	0	0	139	100.00
3	Birthdate		D	S	0				0	0	11	0	0	139	100.00
4	Race		A	S	0				0	0	9	0	0	138	90.28
5	Physician	Referral physician	A	S	0				0	0	29	0	0	139	100.00
6	Address		A	S	0				0	0	56	0	0	137	98.56
7	Phone		A	S	0				0	0	12	0	0	65	46.76
8	DT1	First visit DT	D	S	0			yes	0	0	11	0	0	139	100.00
9	Symptoms	Symptoms and signs	A	M	0				0	0	24	0	8	139	100.00
10	Dur1	Dur of symptoms and signs	I	M	0				0	0	3	0	16	112	80.58
11	Status	Hormone status	A	M	0				0	0	20	0	7	139	100.00
12	DT2	Hormone DTs	D	M	0				0	0	11	0	3	65	46.76
13	Estrogen	Estrogen type	A	S	0				0	0	2	0	0	135	97.12
14	Family	Family history	A	H	0				0	0	20	0	5	102	73.38
15	Past	Past medical history	A	M	0				0	0	21	0	9	135	97.12
16	DT3	IVP DT (Pre Rx)	D	M	0				0	0	11	0	2	121	87.05
17	IVP	IVP-Roentgen studies	A	M	0				0	0	26	0	2	139	100.00
18	Chest	Chest X-ray	A	M	0				0	0	32	0	13	139	100.00
19	DT4	DTs – Chest, liver, brain	D	M	0				0	0	11	0	4	127	91.37
20	Scans	Liver and brain scans	A	M	0				0	0	1	0	5	139	100.00
21	Laboratory	Laboratory – Pre Rx	A	M	0				0	0	1	0	9	114	82.01
22	Recent	Most rec. Dx evaluation	A	M	0				0	0	10	0	2	139	100.00
23	Long	Long bones	A	M	0				0	0	6	0	2	12	8.63
24	Cervical	Cervical spine	A	M	0				0	0	4	0	2	13	9.35
25	Thoracic	Thoracic spine	A	M	0				0	0	4	0	2	17	12.23
26	Lumbar	Lumbar spine	A	M	0				0	0	16	0	2	27	19.42
27	Sacrum	Sacrum	A	M	0				0	0	4	0	2	23	16.55
28	RInnominate	R. innominate bone	A	M	0				0	0	4	0	2	18	12.95
29	LInnominate	L. innominate bone	A	M	0				0	0	4	0	2	19	13.67
30	Pubis	Pubis	A	M	0				0	0	4	0	2	19	13.67
31	Skull	Skull	A	M	0				0	0	4	0	2	19	13.67
32	RIlium	R. ilium	A	M	0				0	0	14	0	2	22	15.83
33	LIlium	L. ilium	A	M	0				0	0	10	0	2	14	10.07
34	Ribs	Ribs	A	M	0				0	0	4	0	2	9	6.47

Fig. 14.

file. At the present time there are 4067 records existing in the file. Therefore, DIRAC prompts the user for the 4068th record. The user types out the information and terminates his update with 'END'.

3.3 *The radio-therapy clinic*

3.3.1 Statement of problem. The radio-therapy clinic at Stanford Hospital has undertaken a study on 400 patients who have incurred prostate cancer. These patients have been followed-up for approximately 3–8 years and the data which has been amassed in each patient file is considerable. It is not the object here to draw conclusion on the study per se (this is being done in the medical journals), but to present the reason for which the file was created and implemented under DIRAC. Specifically, to obtain results in this study, various subsets of record information are needed to be retrieved and then correlated either against time or against each other. To do this by hand would be useless. To write a dedicated computer program to retrieve these subsets and correlate them would solve the problem partially. However, this method limits the investigators to before-the-fact analysis, and provides no procedure by which statistics and correlations other than those that were predetermined and programmed can be obtained. Thus, the investigator is faced with a reprogramming task to answer those questions that were not anticipated. DIRAC provides a solution to this problem by allowing the user to browse his file and interrogate his file in a non-procedural mode not having to anticipate questions that might arise.

3.3.2 The prostate cancer file. Figure 14 is a status report of the prostate cancer file. Note that this file contains a large number of fields due to the large

Stanford University Computation Center	Status report for file M300 28 September 1970	(Class 1)	Language: DIRAC-1A

Description: File of prostate cancer patients

Creation date: 16 September 1970	1 = Type	File name:	Prostate
Record notation: SN$	2 = Multiplicity	File created by:	Ray
Field notation: #F	3 = Indexing	Record length:	1024
Disposition: Public	4 = Code residence	No. of fields:	68
No. of records: 139	5 = Code type	Latest update on:	16 September 1970

Field identification			Statistics and validation information												
Field name		Description	Storage					Validations			Statistics			Existence	
			1	2	3	4	5	Nec.	Size	Sub.	Size	Dec.	Sub.	Rec.	PCT (%)
35	Pelvis	Bony pelvis – other	A	M	0				0	0	4	0	2	11	7.91
36	Bones	Bones – other	A	M	0				0	0	16	0	2	8	5.76
37	DT5	Diagnosis DT	D	S	0				0	0	11	0	0	134	96.40
38	Method	Dx method	A	M	0				0	0	10	0	8	139	100.00
39	Cardiovascular	Physical examination	A	M	0				0	0	11	0	3	139	100.00
40	GU	GU-prostate, phys. exam.	A	M	0				0	0	14	0	8	139	100.00
41	Lymph	Lymph nodes	A	M	0				0	0	8	0	4	139	100.00
42	Leg	Leg edema	A	M	0				0	0	2	0	3	139	100.00
43	Staging	Staging summary	A	M	0				0	0	8	0	5	139	100.00
44	PreviousRx	Previous Rx	A	S	0				0	0	11	0	0	138	99.28
45	Therapeutic	Therapeutic plan	A	M	0				0	0	1	0	2	137	98.56
46	Bladder	Bladder infection	A	M	0				0	0	3	0	2	139	100.00
47	Histopathology	Histopathology	A	M	0				0	0	32	0	11	139	100.00
48	Radiation	Radiation	A	M	0				0	0	26	0	13	138	99.28
49	Area	Area in CM	I	M	0				0	0	4	0	6	136	97.84
50	DT6	DT of 1st Rx	D	S	0			yes	0	0	11	0	0	139	100.00
51	DT7	DT of last Rx	D	S	0				0	0	11	0	0	136	97.84
52	Size	Size of gland	A	M	0				0	0	1	0	2	136	97.84
53	BonyMets	Bony mets	A	M	0				0	0	2	0	4	35	25.18
54	Dur2	Leg edema dur	A	M	0				0	0	2	0	3	6	4.32
55	Complication	Complications	A	M	0				0	0	65	0	13	135	97.12
56	Dur3	Dur of complications	I	M	0				0	0	3	0	12	80	57.55
57	Impotency	Impotency	A	S	0				0	0	2	0	0	49	35.25
58	DT8	DT of last follow-up	D	S	0				0	0	11	0	0	137	98.56
59	Follow	Follow-up status	A	M	0				0	0	28	0	4	137	98.56
60	Failure	First failure	A	S	0				0	0	1	0	0	134	96.40
61	DT9	DT of first failure	D	S	0				0	0	11	0	0	45	32.37
62	Site	Site	A	M	0				0	0	14	0	9	47	33.81
63	DT10	DT of liver mets	D	S	0				0	0	11	0	0	1	0.72
64	Failure Rx	Rx of failures	A	S	0				0	0	4	0	0	112	80.58
65	Autopsy	Autopsy	A	M	0				0	0	35	0	7	137	98.56
66	DT11	DT of brain mets	D	S	0				0	0	11	0	0	10	7.19
67	Remarks	General comments	A	S	0				0	0	130	0	0	46	33.09
68	Cause	Cause of death	A	S	0				0	0	63	0	0	36	25.90

Fig. 14 (Continuation).

```
1.    FILE NAME
      :   M300
2.    FILE 'DESCRIPTION'
      :   'File of Prostate Cancer Patients'
3.    DISPOSITION (PUBLIC/PRIVATE)
      :   PUBLIC
          ENTER 'AUTHORIZED UPDATE USERS'
      :   'Ray'
          ____________
          REVISIONS ?
      :   REVISE 3
3.    DISPOSITION (PUBLIC/PRIVATE)
      :   PRIVATE
          ENTER 'AUTHORIZED QUERY USERS'
          'Ray Bagshaw'
      :   ENTER 'AUTHORIZED UPDATE USERS'
          'Ray Bagshaw'
          ____________
          REVISIONS ?
      :   NONE
            .
            .
            .
```

Fig. 15. Example of create mode revision process.

amount of diagnostic information obtained from the patients. Note also that there are a large amounts of 'DATE' fields, thus allowing the user (the doctors in this case) to interrogate the file with respect to date or time limitation (this aspect is further explained in the next section). This file is unique from the other files discussed in this paper for the following reasons: 1. Field specifications are numerous. 2. All modes of information are utilized except REAL (CODES to be implemented in a later version of DIRAC). 3. The file is static, i.e., it does not change over time.

3.3.3 Interactive creation and revision of files. The environment under which the user can create or revise files or records within files is unique to DIRAC. A typical example of the error recovery and revision procedures that DIRAC allows the user during the creation phase is as follows (Figures 15 and 16):

DIRAC first asks the user to type a 'file name' and then a 'verbal description' of the file which he is creating[1,2]. It then asks the user to designate the disposition of the file, whether it will be public or private[3]. When this section has been completed the user may discover that he forgot to specify that the file be kept private and to place a certain name under AUTHORIZED UPDATE USERS. Thus, he now has the option to revise a specific paragraph as shown in the illustration and to enter the name(s) he wishes.

Each request by DIRAC has associated with it a specific error recovery message. If an error is made either in typing or responding with the wrong type of 'VALUES' then the recovery process is triggered such as is displayed in Figure 16.

```
4.   SPECIAL NOTATION FOR RECORD AND FIELD?
     :  NONE
5.   RECORD LENGTH ?
     :  1024
        REVISIONS ?
     :  REVISE 5
5.   RECORD LENGTH ?
     :  511
        ***!                   ____________53
                               LENGTH MUST BE A POWER OF
                               2, FROM 16 TO 2048 WORDS.
5.   RECORD LENGTH ?
     :  256
        REVISIONS ?
     :  NONE
6.   SUPPLY NAME AND 'DESCRIPTION' FOR ALL THE
        FIELDS
#1 ?
     :  Name 'Name of Donor'
              .
              .
              .
#33 ?
     :  Comments 'General Remarks'
#34 ?
     :  END
        REVISIONS ?
     :  NONE
7.   SUPPLY DATA TYPE AND MULTIPLICITY
     :  ALPHA SINGLE #1 #2 #4 #5 #6 #13 . . . . .
     :  DATE SINGLE #3 #8 #37 #50 #51. . . . . . .
     :  INTEGER MULTIPLE #10 #49
     :  ALPHA MULTIPLE #15 #30 #33
     :  DATE MULTIPLE #12 #16 . . . . . . . . . . .
8.   VALIDATION
     :  #1 NECESSARY
     :  #2 NECESSARY
     :  #18 SUBFIELDS 13
     :  #50 SIZE 1
     :  #9 NECESSARY
     :  END
        REVISIONS ?
     :  NONE
        ****!                  ____________51
                               PROPER RESPONSE IS EITHER
                               'NONE' OR 'REVISE'
        REVISIONS ?
     :  NONE
THE FILE HAS BEEN CREATED.
```

Fig. 16. Example of create mode error recovery process. (Con't from previous example.)

It must be remembered that during the create phase of DIRAC field information for each record is not input into the file. This is done during the UPDATE phase. The CREATE phase is solely concerned with the structure of the file and the parameters which allow the user to define this structure.

4. *Conclusions*

The three applications discussed in this paper are exemplary of the manner by which a file-oriented language might be applied. Even though the applications are very different in nature, they have the same basic structure. In one application – radio-therapy – the files are used primarily for research purposes and afford no administrative or accounting function. The other two applications discussed – Hematology and Blood Bank – utilize interface techniques which provide inventory control as well as file management reports. In other words they are in the truest sense an information system used by medical personnel for research and by management for administration and accounting.

As the time-sharing technique or interactive mode of computing becomes more prominent the flexibility of file-management languages are seen to have special significance in the long-neglected field of medical information processing.

Zusammenfassung. Um den Zugang zum Computer auch für den Nichtfachmann möglichst einfach und effektiv zu gestalten, müssen neue Programmiertechniken entwickelt werden. Ein Teil dieser Neuentwicklungen beschäftigt sich mit dem Aufbau und dem Gebrauch von Computer-Dateien. Zu diesem Zwecke wurde DIRAC an der Stanford-University entwickelt und getestet. Diese Sprache ist die erste einer ganzen Familie von Prototypen, welche eingesetzt werden, um die organisatorischen Probleme der Speicherung und Wiederauffindung wissenschaftlicher Information zu lösen. Die Sprache ist so konstruiert, dass die Benützer via einen Fernschreiber, der weit von der Computer-Zentrale entfernt aufgestellt sein kann, mit Hilfe einfacher Schlüsselwörter ihren Zwecken angepasste Kartotheken aufbauen können («create»), die einmal gespeicherte Information ergänzen, zerstören oder modifizieren können («up date»), die Kartothek nach fast beliebigen Kriterien abfragen und gesuchte Informationen heraussortieren können («query») sowie den Inhalt der ganzen Kartothek herausdrucken und einfach statistisch bearbeiten lassen können («status»). Ein kurzer Überblick über den Sprachaufbau von DIRAC und Beispiele für den Gebrauch von DIRAC durch verschiedene medizinische Abteilungen der Stanford-University sind beschrieben.