EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

PS/CO/Note 84-04
20.1.1984

A DATA BASE FOR THE PS CONTROLS PROJECT

J. Cupérus

Geneva, Switzerland

1. INTRODUCTION TO DATA BASES

Any collection of data, on a computer or in a filing cabinet, can be called a data base. We shall restrict the name to collections of data on a computer which can be updated, searched and printed with a unique set of programs. We can distinguish 3 main types:

1.1 Flat file data bases

The files are composed of fixed-format records (flat files). The files are independent of each other. Some systems allow text fields which are in fact pointers to variable format text records in a sister file. Examples are NORD-IR (Appendix 1) and our home-made INFO (Appendix 2).

1.2 CODASYL data bases

Data are stored in flat files of different formats tied together with pointers. All relations between data must be specified by the data-base manager and this can get quite complex in some cases. Because of this, the data are vulnerable to corruption and adequate precautions must be taken (backup + incremental file). Relations which are not foreseen by the data-base manager cannot be explored by the user. The system needs close supervision and demands much effort from the data-base manager. Changes to the data-base organisation are difficult. In return for all of this, you get fast response. Such data bases may be good for commercial applications with large sets of data in a fairly stable relationship but are badly adapted to our ever-changing requirements. An example is the NORD-SIBAS data-base management system (Appendix 3).

1.3 Relational data bases

The data are again stored in flat files. The relations are defined implicitely with keywords which may be repeated in several files. These keywords can be names or part numbers or whatever. A search is fast if the file is sorted according to the field in which the keyword is and slow if not. Simplicity, reliability and versatility must be paid for by an occasional slow search. ORACLE (Appendix 4) is the relational data base chosen by LEP as being best suited to their purposes.

2. EXISTING DATA BASE FILES

The INFO file system has been in use for several years. Not counting some private ones, it has the following files:

EQ-DATA-BASE

Contains information per equipment (about 2300 entries at present): OB-name, old name, FEC, EM, EQ, tree and working-set information, alarms information, CAMAC addresses, references to SETUP-INFO. Future additions could be completion of the CAMAC addresses and other information for EM initialisation.

SETUP-INFO

The complement of the previous file. Information for groups of equipment (or for the whole EM). The main purpose is to get repetitive information out of EQM-DATA-BASE. Is used at present for setup but could be used in almost the same way for EM initialisation.

Q-CHECK-DESCR

Describes the control/response patterns per group of similar equipment. Used by the Alarms system.

STAT-WORD-DESCR

Describes the status words including the meaning of each bit (= reference to AL-MES-DICT). Used by the Alarms system.

AL-MES-DICT

28 character messages divided in 16 categories: general messages, alarm system messages, status faults, internal faults, external faults, file system errors, NODAL errors, CAMAC errors, EM errors, etc.

DIRECTORY-PROG

List of programs with information for production planning.

3. PROPOSAL FOR NEW DATA BASE FILES

The files necessary to implement the new structures in a data-base environment will be discussed in a separate note. What follows are just a few documentation files which could be implemented right now with a good return for the effort to fill them in.

FILE-DIRECTORY

List of files in use in the PS computer system: system name/ user name/file name/type/computers/function/responsible/archive location + other information if useful. Would e.g. enable to find back all files belonging to a system or loaded on a segment of a FEC.

DOCUMENT-LIST

List of all documentation of interest to the Controls Group: identifaction number/title (shortened)/authors/date/keywords (out of exclusive list)/source.

HANDBOOK

Three levels of index (max. 44 entries per level) plus up to 6 pages of text (max. 22 lines of 80 characters per page). INFO has special facilities for browsing through such a structure. Would contain news, recipes, descriptions, recovery procedures, responsible persons, jokes and dirty tricks.

4. THE RT DATA-BASE

No general purpose data-base system is fast enough for an RT environment. Therefore, a set of files is implemented on the TREES computer. Access to each file is through an index to the main key. If necessary, an index is made for a secondary key (e.g. primary key FEC/EM/EQ and secondary key OB-NAME). The relations between the different files can be exploited by the application programs in a simple way.

The files are derived from the PRDEV data-base with application programs (MAKE-WSET, MAKE-ALARMS) and an INFO facility for dumping the files in the required format with an index. The resulting RT files must be downloaded by FLOPPY on the TREES computer.

The RT data-base is used at present by working sets (and indirectly by most application programs), get-put, alarms, OB-dictionary, setup and certain CVM initialisations.

5. COMPARISON INFO - ORACLE

INFO and ORACLE seem to be the only suitable candidates for our data-base. Let us compare them in some more detail:

General Facilities

ORACLE ia a powerful relational data-base system made for commercial needs. Special features are virtual fields which are the result of a computation involving other fields and good capabilities for statistics on the data. String, integer and real are the only accepted data types.

INFO has limited relational capabilities (for DB-application programs only) and limited statistics. On the other hand, INFO has several special features for downloading the RT data-base (see Appendix 2). It is of course possible to code an octal integer as a string, as necessary in ORACLE, but then the DB-application program has to do the conversion.

Single File Searches

ORACLE is somewhat more general (e.g. it allows comparison of fields with other fields). INFO has all the important features and can search for a match with a pattern or a SINTRAN-like abbreviation anywhere in a string while ORACLE detects only an exact match or a match with the beginning of the string.

Data Safety

Safety of INFO is excellent due to simplicity and independent files. ORACLE is certainly worse in this respect although better than CODASYL data-bases.

Text Fields

INFO supports text fields with up to 22 lines of 80 characters. ORACLE has only fixed length strings of up to 254 characters.

Speed of Operation

Both have reasonable response times. ORACLE can occasionally be very slow but only on applications which would be impossible with INFO anyway. Sorting takes a long time for both systems if the file is long.

Availability

INFO is on PRDEV while ORACLE is on a VAX accessible through INDEX and CERNET.

DB-application Programs

In Pascal for INFO. In Fortran for ORACLE and on a remote computer.

Ease of Use

INFO is very easy to use. ORACLE asks for a few days of training for the general user and much longer for the data-base manager. Maintenance of the system takes more time for ORACLE.

Data Entry

Filling in of new data is straightforward for INFO: select the fields you want to work on, enter the append mode and enter the data with fields separated by commas until all data are filled in. After each CR the user is prompted with the title of the next field to be filled in. In ORACLE you can specify a layout specially adapted to your application but specifying the format takes time. This is fairly typical for the difference between the two systems: INFO gives simple general purpose facilities while ORACLe gives more possibilites but at a price.

6. CONCLUSIONS

Relational capabilities are important if you want interactive response to queries such as: how many gadgets did salesman Jones sell in France in the first quarter of last year? Commercial companies want this capability because their day-to-day operational decisions depend on them. Our main preoccupations are different: give support to the RT data-base and document our system. Both these applications can be done without the relational facilities. Nevertheless, it would be nice to have them but I think it gets interesting for us only when we can get ORACLE on a NORD computer and with a PASCAL interface.

In the meantime, it would be interesting to get some hands-on experience with ORACLE. I propose to open an account on the VAX and put one or more files of an administrative character on it. DIRECTORY-PROG would be a good file to start with.

In parallel I suggest to expand our present INFO system with new files. The data are just as easy to type into INFO as in ORACLE and it is fairly straightforward to transfer the data in block from one system to the other. A year or so from now we will probably have a better idea of the way to go.

Appendix 1 NOTIS-IR (Information Retrieval)

In fact a card index on computer [1]. On the "card" you can make a nice layout of index fields and text fields with a total maximum length of 200 lines which makes it possible to store a document (letter, note, etc..) with the index to retrieve it.

The index fields (max. 60) can be used to create a flat file data-base with rather extensive facilities for searching and printout.

Storage requirements are about 2 times the actual information content for long files and much more for short ones. Updating and searching the data-base is rather easy but creating, checking and extending the length of the files is specialist work. Due to many auxiliary files with pointers the data-base must be rather sensitive to crashes and recovery is a difficult procedure which works only if sufficient precautions are taken by the data-base manager.

Three shortcomings make the system unsuitable for our purposes:

- The format of the file cannot be changed (adding, deleting or changing fields) without losing the data;
- nothing seems to be foreseen for interaction of external programs with the data base;
- block transfer of data from or to an independent flat file is not possible.

Appendix 2 INFO

A home-made flat file system [2] with fairly powerful facilities for creation, searching, updating, sorting, file structure modification, printing and external datablock input/output. It allows free-text fields (limited to 22 lines per field, so you cannot write more than a terminal screen can show). Interaction is very simple: it is completely menu-driven and a HELP facility is available in French and English. After a few minutes you can create, update, etc.. you own files and this without ever opening a manual. Editing is according to PED (in the new version which will be released shortly). The program is well structured and easy to maintain.

Downloading of the data to the RT data-base is facilitated by some data types which are not normally found on other data bases:

- an associated codeword file allows symbols (e.g. property codes) to be written and displayed in letters but coded in integers;

- octal, binary and hex types are allowed;

- up to 8 integers can be packed in a single word according to the specifications of the user (e.g. CAMAC addresses LCN written as 2/23/15).

- a list of integers or codewords can be defined as a type occupying a single field (e.g. working set list for a machine variable or property sequence for setup); this solves part of the problem of the 1 to many relationship in data-bases.

- INFO can dump a file + index in a format suitable for fast RT access.

The new max. number of fields is 43 (due to space limitations in the 1 page header) but the effective number is much higher because a list is counted as one field.

DB-application programs can be either mode files or Pascal programs with a simple interface to the data-base. A Pascal program can have several files open at the same time. This introduces part of the power of relational data-bases through the back door.

Appendix 3 ACCESS + SIBAS

ACCESS [3] is a NORD process which provides interaction of the user with the data-base: flat files, ISAM files or SIBAS data-base; but for SIBAS [4] it allows reading only, for writing on the data-base you must use another process. The whole thing is really very complex with a set of interlocking programs each trying to do different things in different configurations.

The data-base is of the CODASYL type which should give fast response in a well-defined environment but is difficult to create and modify and not at all adapted to our ever-changing environment. Data safety is probably low due to the monolytic character of CODASYL data-bases.

The only group in CERN which has tried SIBAS seems to be UA1 but they abandoned it again as being unpractical.

Appendix 4 ORACLE

ORACLE [5] is a relational data-base and was chosen by LEP as being best suited to their needs. It is available on VAX and IBM computers. In CERN it is implemented by DD on a VAX computer, accessible by INDEX and by CERNET for file transfer. The system is rather popular and the VAX is at present overloaded. The query language is SQL, standardised by IBM. Pascal is not implemented, at present, for applications programs, but Fortran is allowed.

Relational data-bases are rather recent in design. DD has the system operational for more than a year and most of the initial troubles seem to be solved. NORD seems at present not very interested in implementing the system but this may change under user pressure.

A comparison with INFO is made in §5 of this note.

REFERENCES and RELATED PAPERS

[1] NOTIS-IR User's Guide, NORD publication ND-63.005.01, December 1982

[2] The INFO File System, J. Cupérus, PS/CO/Note 80-2, 25.1.1980. An updated version of this note will be circulated soon.

[3] ACCESS User Guide, NORD publication ND-60.153.02, January 1983.

[4] SIBAS Users Manual, NORD publication ND-60.127.

[5] ORACLE Users Guide, Relational Software Incorporated (RSI), publication RSI-100.

[6] Proposition d'organisation de la documentation des projets du groupe PS/CO, A. Gagnaire, PS/CO/WP 82-35, 22.8.1982.

[7] Comments on [6]. R. Cailliau, PS/CO/WP 83-102, 1.8.1983

[8] Comments on [6]. J. Cupérus, PS/CO/WP 83-104, 4.8.1983

[9] Documentation Computer on possible NOTIS satellite network (draft for comments), B. Kuiper, PS/CO/WP 83-98, 27.7.1983.

[10] Office Automation at PS (draft for comments), R. Cailliau, 18.7.1983.

[11] Data bases in the LEP control system, M.C. Crowley-Milling, LEP Note 375, 6.5.1982.

[12] A data-base as a bridge between hardware and software, H.W. Atherton, M. Rabany, R. Saban, CERN/SPS/82-13 (ELE), 26.8.1982 (presented at 5th IASTED International Symposium on Measurement and Control).

[13] Data-bases for LEP Controls, M.C. Crowley-Milling, LEP Controls Note 42, LEP Note 476, 25.10.1983.

[14] The presented state and future requirements of the LEP data-base project, J. Schinzel, Memo to ORACLE users, Nov. 1983.

[15] Experience with the ORACLE data-base management system on a VAX, P. Benassi et al (paper presented at DECUS Europe, Zürich 1983.

Distribution
CO list 7a)

List 7a) Software Information

V. Adorni. G. Baribaud, G P Benincasa M Bennett,
B. Bobbio, R. Brown, P. Burla, R. Cailliau, B. Carpenter.
L. Casalegno, V. Chohan, G. Cuisinier, J. Cupérus,
A. Daneels, G. Daems, R. Debordes, Ch. Dehavay,
R. Delgado, F. Di Maio, A. Gagnaire, F. Giudici,
W. Heinze, P. Heymans, B. Kuiper, M. Lelaizant,
J. Lewis, E. Malandain, M. Martini, P. Martucci,
L. Mérard, K. Osen, F. Perriollat, R. Pluta, C.
C. Poinard, J.P. Potier, G. Quickfall, W. Remmer,
J. Redard, Ch. Serre, C.H. Sicard, P. Skarek,
Ch. Steinbach, N. Vogt-Nilsen