5

COMPUTER–ASSISTED NEONATAL DATA BASE DESIGNED PRIMARILY FOR CLINICAL RESEARCH

Daniel P. Lindstrom,
Robert B. Cotton

INTRODUCTION

There is great diversity among the users and uses of perinatal data systems. The experiences of given users cannot be fully appreciated or well understood without knowledge of their basic perspective. At Vanderbilt University, our computer-assisted neonatal data base developed from a need to plan and carry out clinical research projects. The hardware was purchased with research funds, and the primary focus of application continues to be research, although valuable spinoffs to other applications have developed.

Initially our computer-assisted neonatal data base was implemented via a large time-sharing system. The several programs required for data input, editing, and retrieval were written by us in FORTRAN. However, we were soon plagued by slow response time and the parkinsonian propensity of data bases to overflow their allotted disk space. In 1977, we obtained grant support to purchase our own laboratory minicomputer system, consisting of a Digital Equipment Corporation PDP-11/34 with 56 kilobyte (KB) memory, 12.5 megabyte (MB) disk, VT-11 vector display, A/D converter, and Printronix 300 line printer/plotter. The operating system used is the RT-11 V3B.

Since our existing data base software had to be rewritten for the new computer, we were able to incorporate modifications suggested by our previous experience. The resulting software is tailored to our research needs, yet is flexible enough to accommodate other applications as they arise. Adaptations of the software are currently in use at several medical facilities in Gothenburg, Sweden.

This chapter explores in considerable depth the design features and software structure of the Vanderbilt University computer-assisted neonatal data base, with its primary focus on the performance of clinical research projects.

DESIGN CRITERIA

The design criteria for our neonatal data base system included the following:

Convenience of use
Flexibility and ease of modification
Efficiency of use of disk space, memory, and computer time
Ability to collect and plot time-oriented data
Flexible selection and retrieval of data
Ability to generate standardized reports
Ability to process data through statistical analysis programs

OVERVIEW OF DATA BASE STRUCTURE AND CAPABILITIES OF THE SYSTEM

The structure of the Vanderbilt data base is very simple, comprising a root data file and subsidiary files for collection of timed variables. The root data file, in turn, is composed of fixed-length keyed records. This structure suffices for our neonatal intensive care unit (NICU), since only one hospitalization occurs for each patient. Key numbers are normally assigned sequentially by the computer when the patient is admitted to the NICU and are written on all paper forms used to collect data. The root data file contains most of the data, including patient identification information, maternal data, physical examination findings, initial laboratory test values, diagnoses, and outcome.

The subsidiary files are for collection of timed observations of sets of variables, such as sequential blood gas values and ventilator settings. These serial data files are created on request and expanded as necessary to accommodate new data.

This data system enables the user to define the content of the patient record that will comprise the neonatal data base. The patient record is refined by answering a list of questions relevant to the medical, investigational, or administrative needs of the user. To provide flexibility in the kinds of data to be stored, answers to the questions can be expressed in any of nine different formats (Table 5.1), including yes/no, multiple choice, date, time, numeric laboratory values, and alphanumeric character strings. The neonatologist can also annotate each question with additional explanatory text that can be used optionally to prompt (cue) the person entering the data.

The list of questions, along with the format of the answers and any annotative text, is used by the program called QUESTN to define the structure of the data records. After this initial preparation, the user can begin to create, store, and edit individual patient records, using the program called DATAIN. While data are being inputted or reviewed, DATAIN displays each question sequentially on the computer terminal. At this point, either a valid answer or a question mark can be entered as a request for a display of explanatory text. When the last question has been entered or a special termination code has been typed, the entire record is stored.

Retrieval of information from the main data base is provided by the program called SELECT, which produces a summary of all patient records that meet selection criteria specified by the user. These selection criteria are specified by designating a value or range of values for a set of the question responses (excluding alpha numeric character data). The logical relationships between the members of the set of responses are specified by use of the logical operators "AND," "OR," or "NOT." In this way, SELECT could be used to summarize the patient records of, for example, patients born between specified dates

Type	Use	Format
Logical	Yes/No/Mild/Mod/Severe	4 bits
Character	Name, etc.	8 bits/character
Integer	Numeric or coded	4, 8, or 16 bits
Scaled integer	Numeric (one or two decimals)	4, 8, or 16 bits
Floating decimal	7-digit precision	32 bits
Bit coded	Multiple answers	4, 8, or 16 bits
Date	Year, month, day	16 bits
Time	Hour, minute, second	16 bits
Relative time	5-minute intervals since birth	16 bits

TABLE 5.1. Nine different question formats used to define the content of the patient record and thus the structure of the individual data bases.

"AND" who weighed within a specified range "AND" who had hyaline membrane disease "OR" group B streptococcal pneumonia but who did "NOT" have intraventricular hemorrhage. Selection criteria can be based on as many as 50 of the DATAIN questions, so that there is no practical limitation of the number of combinations of criteria used to specify a subpopulation of patient records.

SELECT summarizes the specified subpopulation of patients according to outcome (i.e., lived/died) and to birth weight intervals of 250 grams. In addition to this standard summary, up to 50 of the question responses can be designated as output variables, for which the mean, standard deviation, number of valid entries, and range of each output variable will be printed, along with the number of records for which that variable was missing. For example, SELECT could be asked to summarize the Apgar scores, arterial pH values, and duration of hospital stay, in addition to providing the survival table by birth-weight intervals, as well as a list of patients selected. This information is sufficient to compare variables from groups of patients using simple statistical tests.

SPECIFICS OF SOFTWARE STRUCTURE

As can be surmised from the above overview, the software for this neonatal data base system is written in three main parts. The first program, QUESTN, is used to define the data files; the second program, DATAIN, is used to enter and edit the data; and the third program, SELECT, is used to retrieve selected data.

Several other programs plot serial data, list patients included in the data base in order of key or alphabetically by name, produce periodic summaries of the occurrence of specific diseases and mortality rates, and delete entire records, as well. The details of these programs unavoidably depend on the structure of the data base and individual hardware configuration to some extent. However, the software has been written to be as general as possible without compromising its ability to meet our own needs. Nearly all the programs are written in the PDP-11 implementation of FORTRAN IV, and the source code is heavily commented to aid in understanding software.

QUESTN—Data Base Definition Program

In initiating a data base, QUESTN must be used to define the structure of the data file. Input to this program includes the text of each question to be answered during the data input procedure, a coded number representing the type of data to be stored, and optional lines of text that further explain both the question and the valid responses to the user. The program then assigns storage for each field of data so as to make the resulting data record as compact as possible.

A file containing the question text and the information needed to encode the data is then generated by QUESTN. This file can be thought of as a directory that maps each question into its corresponding answer, and vice versa. A second file contains the optional lines of explanatory text that can be recalled during data input by typing a question mark in response to a question. A third file containing such information as the size of the data record, the creation date of the question file, and an array of initial values for the data record is also created.

As mentioned previously in the overview of the data base structure, the two major types of files are the root data file and subsidiary time-sequenced files. The root data file has one fixed-size record per patient, each of which is keyed with a unique serial number, so that data can be accessed without reading all previous data. Space must be allocated for each sequential key whether it is used or not, so it is not practical to use the hospital identification number for the key. In our application, this root data file contains the name, birth date, hospital number, and 125 other pieces of patient information.

The subsidiary time-sequenced files contain multiple serial observations of one or more parameters that can vary with time. As previously described, we use this type of data base to collect arterial blood gas (ABG) and ventilatory assistance data, which are used for research purposes. Each record in this data file contains a time variable that is stored as the number of 5-minute intervals since the time of birth. Both this relative time and the actual date and time of each observation are displayed when reviewing the data. A serial data file is created only on request, and the file is expanded as more data are added.

These data are retrieved, analyzed, and plotted by means of programs that are dependent on the nature of the data. ABG data are plotted using the program SPLOT. An example of plotted ABG data is given in Figure 5.1c. Several different serial data bases can be created, but all must be related to the root data file through the key number. Since the times associated with each observation are computed from the time of birth, the latter must have been entered into the root data file before a serial data file can be created for a given patient.

If extra space has been reserved within each record, additional data items can easily be added either to the root or to serial data bases, using the QUESTN program. In order to protect the integrity of the existing data base, QUESTN displays a warning if and when the addition of new questions would encroach on previously allocated storage space. In order to store the data in a compact array, several different

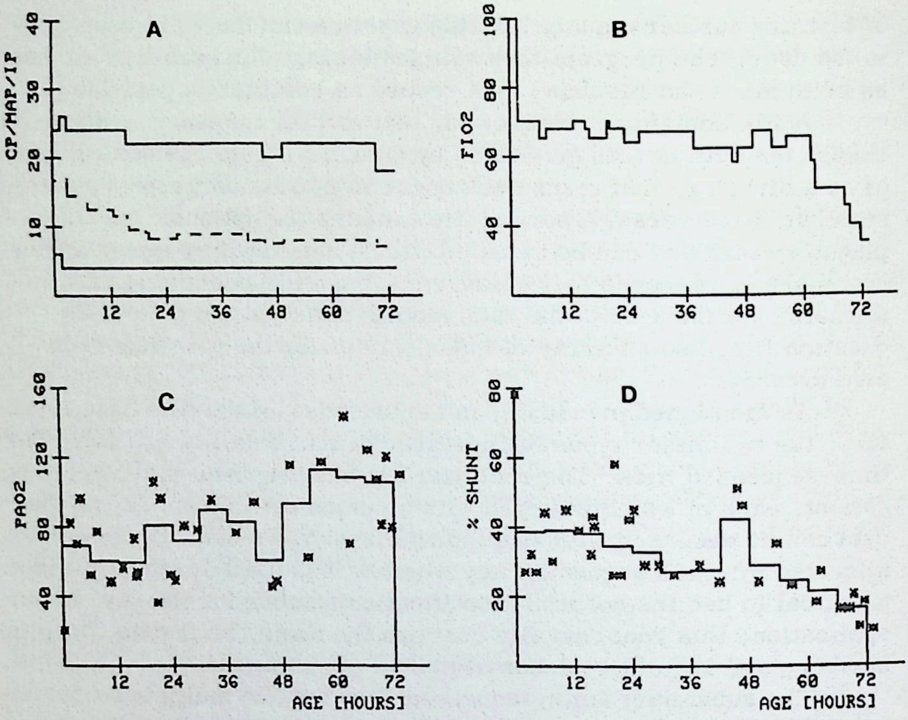

FIGURE 5.1. Direct and derived monitoring data entered by the DATAIN program and plotted using the SPLOT program. (A) Continuous, mean, and intermittent applied airway pressure. (B) Fractional inspired oxygen in percent. (C) Pa_{O_2} values. (D) Percent R → L shunt (venous admixture).

formats are used. The choice of the format depends on the nature of the data to be stored and is specified to the QUESTN program when the data base is marked for future reference (initialized) or modified. For efficient packing of data, the 16-bit PDP-11 words are subdivided into 8-bit bytes and 4-bit "nibbles." The data formats shown in Table 5.1 have sufficed for our applications, but additional formats could be defined as required.

The data base structure is planned according to the appropriate data format, chosen from the options given in Table 5.1. A single line of text must then be chosen—the question that will appear on the video-display terminal during data input. If further explanation of the question is required, any number of optional text lines can be added. The above data are entered into a computer file, using the appropriate text editor. The QUESTN program then reads this text file and assigns each data item to a specific place in the data array. Thus a listing of the entire data base is produced, indicating the location of each item included in the data array. A new text file is then created, similar to the original text file, but containing pointers as to the location of each item in the data record. If any modifications are made to the data base, this text file can be edited and subsequently used as input to QUESTN. Three files containing the complete data base definition, question text, and explanation text are then created by QUESTN. These files must be copied to the disk, which will actually contain the data base, since the files are used by programs accessing the data base.

The final step in establishing a data base is to reserve space on the disk volume for the root data file. The size of this file naturally depends on the length of each record and the number of records that comprise the data base. For example, our neonatal data base of 128 data items uses records of 200 bytes (of which 55 bytes are reserved for expansion), so that 6400 records can be contained in 2500 disk blocks of 512 bytes each. The present software requires that the entire root data base reside in a single file. Hence, a disk of about 2-MB capacity will suffice for a typical NICU. This capacity would allow space for question text files and for serial data files for about 1000 patients. Serial data files can be retired to another storage device to make room for new data.

DATAIN—Data Input Program

Any data-entry program must be convenient for the user. Otherwise, the computer will be regarded with hostility and the utility of tne entire data collection project could be jeopardized. Every effort has been made at Vanderbilt University to develop software that is convenient to work with, efficient, and logical, from the user's point of view. Taking into account the tendencies of experienced users of a computer program to be annoyed by verbosity and of beginners to require detailed explanations of available options, the key is to provide a well-planned question-response format. For example, when an option is requested by DATAIN, the typing of a simple question mark will result in a listing of valid responses.

The DATAIN program is used to enter information into the data files, as well as edit and inspect existing records. It begins by asking for the type of data file to be accessed. Entering "S" indicates a serial data file, and a carriage return alone indicates that the root data file is to be used. In either case, the program asks for the patient's key number. If a question mark is entered, a single line of identifying information will be displayed for the 10 most recent keys, and a key number will again be requested. If a valid key is entered for an existing data record, that record will be recalled. If no key or the key of an empty record is entered, the program will ask whether you wish to create a new record.

If DATAIN is accessing the root data file and creation of a new record has been requested, all the questions will be asked in sequential order. The user can jump to any question number by typing the desired question number preceded by a slash (/). Each data record is marked for future reference if it contains questionable or missing data; answering a question with a dollar sign ($) will replace the existing answer with the missing data code. Typing a question mark will result in the display of any explanatory text for that question that has been stored by the QUESTN program. To store the record at any point in the question sequence, a CTRL-X character must be entered, and CTRL-C will result in immediate exit from DATAIN without updating the current record.

When retrieving an existing record of any data file, the user can select one of four reviewing modes:

1. List all questions and answers on the terminal.
2. Print all questions and answers on the printer.
3. Pause at missing answers for entry of data.
4. List questions and answers, pausing after each for possible editing.

Serial data files are treated somewhat differently by DATAIN, since they are created on the disk only when requested and are automatically expanded to accommodate any number of timed, sequential observations of a set of variables. If no serial data file exists, the user is asked whether one should be created. Preexisting files are opened, and the user is given the following options:

N: New serial data record. All questions will be asked, and a new record will be created.

O: Old serial data records will be recalled for editing or review in reverse chronologic order. After displaying some identifying information for a record, you are asked whether you wish to edit the record, list the data, enter missing data, or return to the start of the program. You can also skip directly to a specified record number or recall the next record, leaving the current one unchanged. These functions are similar to those available for the root data file, as described above.

P: Print a summary of serial data for the current patient, either on the console terminal or on a hard-copy device.

D: Delete a specified serial record. The sequence number of the record to be deleted is entered, the program displays the date and time associated with that record, and verification is requested before the record is actually deleted.

Serial data can be entered in any order, but a request to list or print a summary of serial data will cause the data to be sorted in time sequence. The first question in any serial data record request will be the date and time of the observation. This information is stored as the relative time since birth. The format used for entry of this combined data and time is a 10-digit string of numbers. For example, 2:15 PM on 27 March 1980 would be entered as 8003271415. Dates and times are always checked for validity by the software. The date can be abbreviated by "T" for today's date or by "S" for the same date as that last entered.

SELECT—Data Selection Program

The utility of any data base depends on the ability to retrieve selected data in a useful form. Individual records can be conveniently reviewed by means of the DATAIN program, but this can be a very time-consuming process for compiling statistics on specific subsets of the data. The SELECT program provides a convenient way of selecting subpopulations according to specific criteria and of summarizing data from the selected group. Computer files of selected variables can optionally be generated for analysis using standard statistical software packages. Although the structure of the data base is such that data can easily be accessed by specialized programs for specific applications, most data-retrieval tasks can be accomplished using the SELECT program. Approximately 1000 patient records can be scanned per minute, requiring only one pass through the data, regardless of the complexity of the selection rule. At this point, only the root data base can be accessed by SELECT, and serial data are summarized and plotted using software specific to the nature of the serial data.

To specify the data to be selected, you begin by entering a list of logical terms. These terms are simply the number of the question followed by the minimum and maximum values to be selected. The computer assigns a reference number to each term. After all the relevent logical terms have been defined, the logical relationships between the terms must be specified. The logical functions available are "AND," "OR," and "NOT." As an example, the Boolean expression:

([1 and 2]) or ([1 or (not 3)])

can be entered as 1, 2, AND, 1, 3, NOT, OR, OR. The format of these expressions is called "reverse Polish notation," which allows for very complex expressions without the use of parentheses. The above input for SELECT can involve input from the terminal or it can be recalled from a disk file.

A list of question numbers corresponding to the data to be tabulated can then be specified. The data will be tabulated with means, standard deviations, extremes, number of valid cases, and number of missing cases. Data can also be displayed in histogram format. If desired, the data can be saved in a named disk file for further statistical analysis. A mortality table of selected records showing outcome by birth-weight intervals can also be printed. A final option is to print a list of select patients. This list contains enough identifying information to locate hospital charts or other data on each of the patients.

Implementation Notes

All the above programs were written in DEC PDP-11 FORTRAN IV, except for a few small subroutines written in Assembler language. No effort was made to avoid using nonstandard features of the language. The software should execute properly on any PDP-11-based system. About 2100 lines of code comprise the three programs. The operating system under which this software operates is RT-11 V3B, either single-job or background-foreground monitor, and the programs can be partitioned into overlays small enough that they can be executed in a small time-sharing system called TSX (from S & H Computer Leasing). In nonoverlaid form, each of the programs occupies less than 35 KB of memory. The computer for which the software was designed is a DECLAB PDP-11/34 with 56 KB of usable memory and 12.5 MB of disk storage. All the data base software plus our 6500-patient root data file will fit onto a single 2.5-MB disk.

DISCUSSION

The neonatal data base system has been in operation since 1977 in its present form. By fall 1980, more than 4400 infant records had been stored in the root data file. Essentially complete data (128 items) are entered for any infant admitted weighing less than 1500 g or in whom hyaline membrane disease develops. An abbreviated record of 20 items is entered for all other admissions to the NICU. Total admissions to our unit are more than 900 per year. Time for entry of data: only about 30 min/day.

Changes in our NICU population, including demographic factors, incidence of different diseases, mortality rates, and length of stay, are identified through a compilation of quarterly and annual reports, using the data base and a specialized report generation program. These data are invaluable for long- and short-range planning, for evaluating changes in the management of critically ill infants, for generating reports required by government agencies, and as a teaching resource.

Quality control and consistency are two important benefits of maintaining a specialized data base within a given unit. Several printed forms are used to organize the data in the order and format required by DATAIN. Most of the information is entered soon after admission, and the diagnosis, treatment, and outcome information are entered after discharge. One research assistant is responsible for all data collection and entry. Medical and research personnel who have no previous computer experience find the programs easy to use with a minimum of instruction.

The way in which the QUESTN program maps the text of a question into the part of the data record containing the corresponding data item provides great flexibility in tailoring the data base for a specific application. The basic DATAIN and SELECT programs are specific to our NICU only in the use of certain identifying information, such as name, date and time of birth, birth weight, sex, and gestational age. The portion of DATAIN that maintains serial data files is also specific for our applications, but it can be reprogrammed as required. Such specialized application software as the quarterly report program is very specialized and would probably require extensive modification if used in another NICU.

When this particular data base project began, generalized data base management systems and versatile medical information software, such as MUMPS (Massachusetts General Hospital utility multiprogramming system), were not available for small computers. Currently available software and hardware would probably offer advantages in flexibility, portability, and maintainability over a user-written system such as ours. However, given our needs and our hardware and software constraints, our neonatal data base system has proved a worthwhile investment in resources.

FUTURE DIRECTIONS

The neonatal data base system just described, and the signal analysis and modeling projects currently under way at Vanderbilt University (described in Chapter 18), are beginning to strain the resources of our computer system. One solution is to switch to a true multiprocessing operating system, such as RSX-11M. In such an environment, signal analysis programs would still essentially lock out all other users, at least during actual digitization of analog data. A small satellite computer could easily take care of the signal analysis tasks, interrupting the host computer periodically to transfer partially or fully processed data. Such a satellite computer, or possibly a third microcomputer, would be able to do modeling calculations, so that these techniques would have a greater impact on improving patient care and education of the front-line medical personnel. During the next few years, we plan to move toward making better use of the data we are in the process of collecting, as well as to incorporate additional prenatal and maternal information into our studies.

ACKNOWLEDGMENT

This work was supported in part by a National Heart and Lung Institute SCOR grant (HL-14214).