EXPERIENCE USING STANDARD SOFTWARE TO PROVIDE A COMPREHENSIVE INFORMATION PROCESSING SYSTEM FOR A LARGE AND COMPLEX CLINICAL STUDY

by John M. Long and Joseph R. Brashear

Hyperlipidemia Program
Minneapolis, Minnesota

ABSTRACT

In this paper, we describe how we used a standard database management system (System 2000) and a computer utility to build a sophisticated medical records system in support of a national multi-clinic clinical trial. Privacy, protocol adherence, quality control and other key elements of an ethical clinical trial were satisfied at a fraction of the development cost for the more traditional approach of building a customized system. We feel that old lessons learned in other areas regarding the balance of manual to automated systems and the use of standard software are applicable to medical information systems. Neither a medical setting nor a clinical experiment changes the basic issues of good systems design. Operating and efficiency statistics gathered during two years of system production operation are presented.

ARTICLE

In this paper we will describe our experiences in the development and operation of a data processing system for a national, multi-clinic, controlled clinical trial. It is hoped that the description will demonstrate that controlled clinical trials can produce economical database systems by combining packaged software programs with manual processing techniques and a minimum of customized data programs.

Our clinical trial, funded by a grant from the National Heart, Lung and Blood Institute, is formally known as the Program on the Surgical Control of the Hyperlipidemias, Secondary Prevention Trial (POSCH). We refer to it here as the Hyperlipidemia Clinical Trial or Study.

The Hyperlipidemia Clinical Trial is designed to determine the effect of maximal cholesterol reduction in patients with known atherosclerotic heart disease. One thousand patients are being randomly assigned to either a treatment or control group and then followed for five years. Trial protocol requires a highly restrictive recruiting and screening process to obtain the thousand patients for randomization; we anticipate collecting data from at least 10,000 patients during the screening process. Twenty-two different forms, ranging in length from one to 31 pages, are used for data collection. Obviously both patient management and data management are primary concerns of the trial.

We were additionally faced with a set of environmental constraints. Form design and patient recruiting had already started when computer systems personnel were hired. We expected to be inundated with forms at any moment, before systems and procedures were fully established. And, in the rush to recruit, incomplete attention had been given to the practical day-to-day aspects of how to handle data and analyze the end results of the trial. We were forced to relearn an old lesson: that computer processing represents only one of the major data processing systems areas.

We needed a system that could be operational quickly at reasonable cost. The data processing system is a necessary tool for a clinical trial, but is not the paramount concern and should not become a major expense item. The system also had to be responsive and easily modified. Medical ethics require constant surveillance for unanticipated events and possible procedure termination.

Other constraints on system design were no different than those facing a systems analyst in any application, such as privacy and security of data. Neither a medical setting nor a clinical experiment change the basic issues of good system design. But trial management was faced with a classic computer system dilemma. There are four obvious alternatives in building a data processing system: develop a custom system; share customized hardware and software with other similar applications; develop a generalized hardware-software system specifically for clinical trials, or use a computer utility and available standard software. The Hyperlipidemia Clinical Trial chose the fourth alternative.

We are using a shared CDC Cyber facility at the University of Minnesota and System 2000 as our database management system. Begun in May of 1975, the basic system was fully operational in May of 1976 (that is, we were able to enter clinical data in a controlled operation, provide data accountability, and retrieve reports based on the data). Improvements have continued and we are now

Preparation of this paper was supported by the National Heart, Lung and Blood Institute. 2R10 HL 15265-06 (CLTR)
able to generate all reports very shortly after they have been defined. The basic system was developed in less than one year at a cost of less than $60,000.

A great deal of time has been devoted to gaining input and support regarding reporting formats. The generality of standard software has permitted changes with little disruption to the automated portions of our system. Overall, rather than expend all our resources on computers and software, we have emphasized data content and editing, meaningful data analysis, and effective computer generated reports.

Our medical records database design was based on three primary concerns. First, we needed ready access to large volumes of clinical data. Second, we are required to monitor adherence to the trial protocol by every patient from each clinic. Third, we had to control data flow through a complex editing and certification process. These concerns led to the development of two separate System 2000 databases. The main, or Medical, database contains all of the clinical information collected for every patient. The Administrative database holds the information needed to control data collection and adherence to trial protocol.

These functions are analogous to those found in a traditional circulating library. The Medical database is equivalent to the library stacks, the Administrative database to the card catalog and circulation system.

The Administrative database is on-line and is accessed through a time-sharing terminal. An entry is made for every data form which arrives from a clinic. The database has a tree structure with three hierarchical levels. All data elements, except a text memo, are inverted; that is, they are all key elements and may be used to qualify retrieval or access operations. Complete inversion has given us the capability to retrieve formatted administrative reports with minimal programming effort.

The Medical database, holding the medical records, is also maintained on-line, but is updated periodically in a batch processing mode from punched cards. Although extensive editing and consistency checks are made in the update programs, it is expected that manual control of the input data will make it relatively error free. This database is also structured as a three-level, hierarchical tree, but the structure is unique. Our approach was dictated by storage costs and limitations, as clinical data is submitted on a large number of complex forms. A straight-forward approach would be to define a data set or repeating group with unique data elements for each form. This approach rather quickly leads to realization of System 2000 overhead costs and limitations on numbers of unique elements. The design used was to designate the third level as a two-element data set containing a question number and an answer. Figure 1 is a description of the database structure in System 2000 language.

Figure 1

```
SYSTEM RELEASE NUMBER 2.40P
DATABASE NAME IS MEDDB
DEFINITION NUMBER 58
DATA BASE CYCLE 339
1* REF. NO. (INTEGER NUMBER 9(5))
2* BIRTHDATE (NON-KEY DATE)
3* CL CODE (INTEGER NUMBER 9)
4* SEX (NON-KEY INTEGER NUMBER 9)
5* M.I. DATE (NON-KEY DATE)
6* RAND. GP (NAME X)
7* ELIG. STATUS (NON-KEY INTEGER NUMBER 9)
8* RAND DATE (DATE)
10* FORM INFO (RG)
11* FORM NO. (NAME X(5) IN 10)
12* VISIT NO. (INTEGER NUMBER 99 IN 10)
13* DATE COMPLETED (NON-KEY DATE IN 10)
14* QUES ITEM (HG IN 10)
15* QUES NO. (NAME X(6) IN 14)
16* Y/NO ANSI
17* NUM. ANSI
18* A/W ANSI
19* DATE ANSI

```

One of the principal concerns with using a standard software package such as System 2000 is that hardware costs would become prohibitive as more and more patient data is entered. The POSCH has gathered statistics on mass storage requirements and cost of retrieval as patient records have been added to the Medical database.

Table 1 tabulates the database size in characters at six levels of patient population. A least-squares straight line fitted to the data has a slope of 0.0715 (Y = 0.0715X - 0.335), indicating that each patient will increase the database size by 71,500 characters. This set of data gives us no reason to believe the increment in size for each patient will either increase or decrease as patients are added. However, the study protocol requires the continuing collection of data about each patient for five years. Therefore this method of measurement is too simplistic.

### Table 1

<table>
<thead>
<tr>
<th>NUMBER OF PATIENTS</th>
<th>TOTAL SIZE (MILLION CHARACTERS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>1.29</td>
</tr>
<tr>
<td>41</td>
<td>2.57</td>
</tr>
<tr>
<td>67</td>
<td>4.26</td>
</tr>
<tr>
<td>91</td>
<td>5.28</td>
</tr>
<tr>
<td>116</td>
<td>7.56</td>
</tr>
<tr>
<td>146</td>
<td>10.43</td>
</tr>
</tbody>
</table>
A more interesting measure of storage efficiency would be the cost in characters we are paying to index each item of data we capture. The number of data items can be determined by counting the number of occurrences of element 15 (Question ID) in the third level of the Medical database. We can also determine the number of characters used to represent values of the data items by counting the answer elements in level three and multiplying by the picture size of each answer data element. These two values are tabulated in columns 5 and 3, respectively, of Table II. Subtracting the number of characters for answers from the total number of characters yields a rough measure of the number of characters required to locate a value for any data item in level three. This rough measure is the "Indexing Characters" given in column 4 of Table II.

Dividing the number of indexing characters by the number of data items yields column 6, a measure of storage efficiency in the database. This ratio, indexing characters per data item, has decreased as the number of patients has increased. We suspect that the decrease is attributable to the use of a Unique Value Table in System 2000. As forms and question identification are added, fewer new unique values will be added to the tables. The ratio is expected to continue to decrease, giving some hope that the rate of growth of the database will diminish as patients are added.

The second component of system overhead, cost to retrieve, can be measured directly. The study has used a set of six System 2000 Natural Language statements to measure cost to retrieve:

1) ILO - retrieves the value of one data element from level zero for a single patient.

2) IL1 - retrieves the value of a small set of data elements at level one for one patient at a single visit.

3) IL2 - retrieves the value of one unique data element at level three for one patient from a single form submitted at a single visit.

4-6) GL0, GL1, GL2 - performs the same retrievals as ILO, IL1, IL2 for all patients in the database.

We have tabulated the cost for each of the six patient levels in Table III. Since our project uses a university computer center, absolute costs are not comparable to those of many other users. But the cost differences between patient levels are of interest. These range from little or no difference for ILO and IL1 to 20 percent or more for GL2. The indexing structure of System 2000 results in rapid retrieval of values from level zero with little effect from total database size. Our database structure sacrificed natural language efficiency at the third level for decreased file storage costs. The third level is seldom accessed using natural language commands and is never ac-

```
Table II

<table>
<thead>
<tr>
<th>NUMBER OF PATIENTS</th>
<th>TOTAL SIZE (MILLION CHAR)</th>
<th>CHARACTERS FOR ANSWERS</th>
<th>INDEXING CHARACTERS</th>
<th>TOTAL CHARACTERS FOR ANSWERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>33.9</td>
<td>1.12</td>
<td>1.16</td>
<td>37.752</td>
</tr>
<tr>
<td>41</td>
<td>2.57</td>
<td>1.36</td>
<td>2.33</td>
<td>5.437</td>
</tr>
<tr>
<td>67</td>
<td>4.16</td>
<td>3.14</td>
<td>3.86</td>
<td>8.191</td>
</tr>
<tr>
<td>91</td>
<td>6.78</td>
<td>5.62</td>
<td>5.60</td>
<td>11.962</td>
</tr>
<tr>
<td>116</td>
<td>7.56</td>
<td>7.53</td>
<td>7.57</td>
<td>15.436</td>
</tr>
<tr>
<td>146</td>
<td>10.43</td>
<td>1.035</td>
<td>1.10</td>
<td>11.57</td>
</tr>
</tbody>
</table>

Table III

<table>
<thead>
<tr>
<th>NUMBER OF PATIENTS</th>
<th>DATABASE SIZE (MILLION CHAR)</th>
<th>ILO</th>
<th>IL1</th>
<th>IL2</th>
<th>GLO</th>
<th>GL1</th>
<th>GL2</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>1.29</td>
<td>.09</td>
<td>.11</td>
<td>.45</td>
<td>.13</td>
<td>.40</td>
<td>1.22</td>
</tr>
<tr>
<td>41</td>
<td>2.57</td>
<td>.09</td>
<td>.12</td>
<td>.72</td>
<td>.17</td>
<td>.67</td>
<td>2.42</td>
</tr>
<tr>
<td>67</td>
<td>4.16</td>
<td>.09</td>
<td>.12</td>
<td>1.08</td>
<td>.22</td>
<td>.96</td>
<td>3.94</td>
</tr>
<tr>
<td>91</td>
<td>6.78</td>
<td>.09</td>
<td>.12</td>
<td>1.32</td>
<td>.27</td>
<td>1.66</td>
<td>5.91</td>
</tr>
<tr>
<td>116</td>
<td>7.16</td>
<td>.09</td>
<td>.12</td>
<td>1.63</td>
<td>.32</td>
<td>1.94</td>
<td>7.15</td>
</tr>
<tr>
<td>146</td>
<td>10.43</td>
<td>.09</td>
<td>.13</td>
<td>2.45</td>
<td>.39</td>
<td>2.48</td>
<td>9.87</td>
</tr>
</tbody>
</table>
```