A clinical information system (CIS) for ambulatory care

by CLEMENT McDONALD, BHARAT BHARGAVA and DAVID JERIS
Regenstrief Institute
Indianapolis, Indiana

INTRODUCTION

In this paper we wish to present an evolving ambulatory care information system (CIS) which has been in use by physicians since July 1, 1973. The noteworthy features of this system include:

1. Implementation in a high level interpretive language on a mini computer.
2. Paper based (optically read turn-around documents) rather than electronic terminal I/O.
3. Automated clinical surveillance (the computer works for the physician by searching out errors and danger conditions rather than being a passive repository of data).
4. Data base management orientation to better cope with the constant companion of change.

By ambulatory care we mean medical care provided in physician offices and outpatient clinics. We call the place where such care is given a Care Facility (CF). It might be noted that our host facility is part of a hospital—and to some extent the CIS provides service to the hospital as well as the outpatient clinic.

In the discussion which follows we distinguish between legislative and executive decisions. The former is a decision to change operational rules. The latter decision is a decision to act based on pre-defined operational rules. The purpose of the CIS to be described is to do executive work at each level: the clinical, ancillary and administrative level, and to enlighten legislative decisions by providing a statistical overview of the system’s operation. Because of our emphasis on ambulatory care, the clinical service modules do not serve inpatient care, the ancillary service modules do.

THE CLINICAL INFORMATION SYSTEM

System overview

The CIS is designed to deal with a large, 200,000 active patient population at the administrative-ancillary service level and a smaller 40,000 patient subset of that population at the clinical level. Our limits are based on both mass storage considerations and the statistical realities of our hospital outpatient facility. The largest DEC supported mass storage device holds 43 million words or approximately 168,000 sectors. Ancillary services and administration require access to a fixed minimum of patient identification data. Patient registration requires approximately 100 bytes of storage per patient (depending upon the CF’s needs, more or less storage could be required). Our target of 200,000 patients fits easily and occupies 40,000 sectors; larger populations would be feasible but are not necessary to our environment. Based on our projected average medical record size of 3 sectors, 120,000 sectors allows for 40,000 patients. This limit seems severe in proportion to the size of active registry, however, all clinics do not have equal need for full clinical services. By propitious shifting of patients from active to inactive storage, these limits can be softened. We have not yet taken delivery of our large mass storage device and to date have not accumulated any experience with very large files. Thus we are not certain that the system resources will be able to support all of the activities we have planned for such a large patient population.

Operational elements of an ambulatory care facility

An ambulatory care facility deals with four populations of elements: patients, physicians, clinic and parameters. These elements must be duly registered in the CIS by name and necessary descriptive characteristics. For example, when a patient is registered, we record his name, hospital number, date of birth, race, sex and other characteristics. The data base approach of the CIS allows flexibility in the choice and the number of such attributes.

The notion of a parameter and what should be registered about a parameter may not be obvious. By a parameter we mean any clinical observable, including treatments, tests, physical findings, items of information from the patient’s history, and second order aggregates of preceding elements. Thus, “penicillin” is a parameter, “electrolytes” is a parameter which is second order because it subsumes four primitive parameters: sodium, potassium, chloride and bicarbonate. The parameter set which describes a clinical universe varies from CF to CF. It represents the discrete descriptive approximation to a
continuous universe. For each parameter, its name, synonyms, type (whether tree, scalar, etc.), normal range, absolute range, units, and certain other characteristics must be registered before observations about that parameter can be accepted.

The clinical information system and the care facility

The CIS is designed to serve the CF at three levels: clinical, ancillary and administrative. However, the main impact of CIS on the quality and cost of care is made by the services it provides at the clinical level. The clinical level also involves the most severe problems, hence, the emphasis is on the clinical level in this report.

The inputs to the clinical record include all patient related information. This includes information gathered by the physician, by the nurse, by the lab, by radiology, and by other ancillary services.

One must view separately the problems of trapping ancillary service data from those of physician acquired data. Ancillary service data can be trapped in two modes. The most efficient is via a computerized ancillary module.

The alternative is for a clinic based clerk to enter such data either via optically read forms tailored to the statistical distribution of test usage or via direct terminal entry.

How to trap data produced by the physician during the clinical encounter? This is the thorniest problem. Grossman, et al. have developed an elegant approach involving standardized encounter forms and dictaphone transcriptions. We have taken a different tack. First we assume that most data trapping within the clinical environment will be expensive since it involves in one way or another an expensive resource—the physician. Second, we assume that most of the data he collects is not of sufficient archival importance to justify high costs (for data of short term utility, his handwritten notes will suffice). Given these two assumptions, we let him decide what parameters are of archival importance to him and ask him to record observations for these on our optically read forms. He must distinguish between two classes of importance: conditional or unconditional. In a general medicine clinic, the urine glucose might be designated as a conditionally important parameter under the condition that the patient is diabetic. Conversely, blood pressure would be declared unconditionally important given the proven consequences of hypertension and our ability to reverse them with treatment. Parameters that are of declared importance to the clinician appear on the encounter form (Figure 1). Those that are unconditionally important always appear. Those that are conditionally important appear only when the specified conditions are met. The CARE language to be discussed later provides the mechanism for specifying conditional relationships which vest importance on a given parameter. As one may notice from Figure 1, space limitations constrain us to a maximum of sixteen important parameters on any one patient. This has not been confining for our outpatient environment.

For more detailed data input by the physician, two mechanisms are available. First, special turn-around documents can be tailored to an individual physician's data input needs. As many as sixty different parameters can be specified on a single form. Second, keyboard terminal entry is available. In addition, we are now developing a tree structured multiple choice form which can also be produced as a turn-around document. Our underlying bias is that much of the voluminous free text data found in the conventional medical record is not of sufficient archival value for computer storage, and thus these special input mechanisms will only infrequently be required.

Clinical reports

The first report is the summary report (Figure 2). This report is a flow sheet displaying the time course of all recorded parameters. Physical findings, historical data,
A Clinical Information System (CIS)

**MARION COUNTY GENERAL HOSP.**

<table>
<thead>
<tr>
<th>DATE</th>
<th>CLINIC</th>
<th>PULSE</th>
<th>SYS BP SITTING</th>
<th>DIAS BP SITTING</th>
<th>WEIGHT</th>
<th>FUNCTIONAL CLASS</th>
<th>LEG SWELLING</th>
<th>AM URINE GLU</th>
<th>AM URINE KETONES</th>
<th>DEXTROSTIX</th>
<th>E.R. VISITS</th>
<th>HYPOGLYCEMIA</th>
<th>LOST WORK DAYS</th>
<th># FT LESIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>23-Jan</td>
<td>25</td>
<td>94</td>
<td>120</td>
<td>74</td>
<td>154</td>
<td>3+</td>
<td>0</td>
<td>2+</td>
<td>0</td>
<td>360+</td>
<td>1+</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20-Mar</td>
<td>25</td>
<td>82</td>
<td>152</td>
<td>100+</td>
<td>156</td>
<td>2+</td>
<td>2+</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
<td>2+</td>
<td>2+</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18-Sep</td>
<td>25</td>
<td>80</td>
<td>130</td>
<td>70</td>
<td>148</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10-Oct</td>
<td>25</td>
<td></td>
<td>130</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>04-Dec</td>
<td>25</td>
<td></td>
<td>140</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18-Dec</td>
<td>25</td>
<td></td>
<td></td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>26-Feb</td>
<td>25</td>
<td></td>
<td></td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12-Mar</td>
<td>25</td>
<td></td>
<td></td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 2**

From the collection of the Computer History Museum (www.computerhistory.org)
over time. Change and the rate of change, both of which are crucial determinants in medical decisions, are readily apparent. In addition to the standard total summary, the system provides summaries restricted to a specific time window or a specified set of parameters, the abnormal report which displays only abnormal parameters, the compact report by which parameter trajectories for multiple patients are displayed on a single page. This latter report supports clinical legislative decisions by presenting data about a large number of patients in a compact form. Any of the above formats can be displayed on a CRT terminal in real time or printed on paper in batch mode.

The second report referred to above is the encounter form. We have already alluded to this report which is a two-part turn-around document with many purposes. It supports the physician’s assimilation of the patient’s active treatment states by displaying the patient’s active treatment profile at the top of the form (Figure 1). Notice that the profile is actually a list of prescriptions as written by the physician. This is not by coincidence, since the encounter form is a constituent of the prescription process as we will see later.

The CARE language

Before discussing the third report provided as input to the clinical level, it is necessary to discuss the language by which the physician defines his executive rules. The CARE language provides us with two capabilities, one which helps legislative decisions, the other executive decisions. The language is interpreted by a keyword driven compiler written in BASIC-PLUS. The CARE language allows the user to specify clinical conditions and consequent therapeutic or diagnostic actions in terms of if-then-else constructs. The building block of the language is the parameter clause containing a parameter subject and one or more modifiers. There are two kinds of clauses, action clauses and conditional clauses. For conditional clauses, the modifiers express restrictions on the temporal course of the parameter and the range of magnitudes or changes in magnitudes. Conditional clauses are linked by the Boolean operators “and”, “or”, in addition to temporal connectives “and followed by”, “and preceded by” “and coincident with”. “Then” and “else” can be used to relate conditional clauses to action clauses. The action clause includes a suggestion to do something in relation to the subject parameter and the justification for that suggestion. For example, “order urine glucose because of risk of metabolic toxicity” is a valid action clause.

A purely conditional CARE statement is an inquiry. In response to such an inquiry, the computer will provide the list of patient numbers for which the conditional statement was true. Coupled with the compact summary and a statistical analyzer to be described later, we have a system for supporting legislative decisions in the clinical realm. What is the best treatment for hypertension (high blood pressure)? We can examine our experience to see. How valuable is the serum calcium as a screening test? We can look at the total number of serum calciums, the number that were abnormal and the consequent course of the patient. We can confirm intuitive impressions that a particular disease is too frequently escaping notice or that a particular therapeutic intervention is too often ineffective. We can do retrospective clinical research that has in the past required laborious hand perusal of thousands of pages of medical records.

In summary, the inquiry capability in concert with its two support modules gives us access to the distributed wisdom otherwise locked up in a large number of individual medical records.

CARE statements containing a conditional and an action clause have different functions than those containing only conditional clauses. We call them executive statements because they serve executive functions. A common clinical task is to execute rules of the form: If a given condition is detected, then initiate a given response. An examination of physicians’ orders on any medical ward will reveal statements of the above form written to the nurse and attest to the fact that the physician need not be directly involved in all arcs from condition to response. For diabetics, we see orders written to adjust the insulin dosage according to the level of blood sugar in the urine. For heart patients, the physician writes to adjust the anti-arrhythmic medication according to the number of extra beats demonstrated on the cardiac monitor. For a large number of conditions, he will ask to be called if the blood pressure drops below a specified level. All of these orders are written by the practitioner to be executed by the nursing staff conditioned on the value of one or more parameters. These are called standing orders. We envision a role for the computer analogous but more extensive than that just described—though perhaps not quite as independent. In both inpatient and outpatient CFs, there are multitudinal tasks which can be described in terms of cookbook standing orders. These all involve a detection step and an action step. In general terms, there are many different classes of such tasks: the detection and response to a recent change in a parameter, the detection of a parameter abnormality which influences a drug’s metabolism and the consequent appropriate dosage adjustment, the detection of adverse drug effects by regular checks on indicator parameters, and the initiation of the proper “work-up” of an isolated abnormality. We believe that such tasks constitute a significant burden on the primary care practitioner and are subject to error. By describing such tasks in terms of CARE statements, the burden can be transferred to the computer. It is clear that the computer (at the present) cannot physically accomplish the actions, thus the analogy with standing orders to a nursing service is not perfect. The computer can only produce a list of suggestions which must be executed by a human intermediary. We believe that at least initially the physician should be the intermediary and sieze the suggestions for appropriateness.

To give a feeling of the CARE language without going
SURVEILLANCE REPORT 08-MAR-74

CASE: ELIZABETH 15563--

CONSIDER MEASURING:
URAL [31-OCT-72] TO MONITOR: DYAZIDE
DIAS BP [26-FEB-74] TO MONITOR: DYAZIDE

CAUTIONS:
DIGITOIN
THIAZIDE DIURETIC: INCREASED TOXICITY
PHENOBARBITOL: DECREASED EFFECT
DYAZIDE
BUN= 24 => INCREASED RISK OF METABOLIC TOXICITY
MIGHT REQUIRE CHANGING TREATMENT REGIMEN
URIC= 9.8 & K+ = 5.9 => CAUSE OF METABOLIC TOXICITY
MIGHT REQUIRE D/C'ING
FE SO4
ANTACIDS: DECREASED EFFECT

DO NOT PLACE IN CHART

REFERENCE AVAILABLE THE TWO KEY REFERENCES:
A PRACTICAL GUIDE TO DRUG USAGE... IN RENAL FAILURE
BENNFT ET AL. JAMA 214:1468-1475
DRUG INTERACTIONS
HANSTEN, LEA & FLBIGER, PHIL. 1973

Figure 3

into great detail we present a few examples. Inadequately treated high blood pressure could be addressed by the following CARE statement:

If "diab BP" > 100 then if last "BP meds" > 0 then increase "BP meds" because of under-treatment, else start "BP meds" because of absent treatment.

To declare that a parameter is important conditionally we write the following:

If "diabetes meds" then order "urine glucose"

The application of these rules to a CF is accomplished as follows: The practitioner develops the set of CARE rules which reflect the needs of his own practice. These are used to analyze his patients in a process we call surveillance. Surveillance can be initiated by a patient visit or at regular intervals by the passage of time. At each surveillance the patient's record is tested against the entire set of CARE statements. For a given patient, if the conditions of an executive statement are satisfied, two types of information are saved: (1) the time and value of the specific parameters which are satisfied; and (2) the consequent action clause. Two reports reflect this information, the encounter form and the surveillance report. Test suggestions appear in the "orders" area of the encounter form to facilitate the ordering of that test as shown in Figure 1. Conditionally important parameters whose conditions are met appear in the "observations" section to facilitate their entry into the system.
The surveillance report is best explained by example. Figure 3 shows such a report based on a limited set of CARE statements which relate to drug usage. Drug monitors are suggested at the top of the report, drug interaction and test results with important therapeutic consequences appear under the caution heading. One might notice from the flow sheet for this patient (Figure 2) that the recommendation to discontinue Dyazide was heeded.

In addition to the support of executive decisions, the system assists the actual prescription writing process. The conventional prescribing function requires that the patient’s name and address and the prescribing instructions be written once on an individual slip for each active medication. With the encounter form, only new or changed medications require writing. For maintenance medications to be continued, the physician marks a single bubble and initializes the preprinted prescription. Only drugs under the controlled substance act such as narcotics require a separate prescription. This exception causes little inconvenience since they are rarely prescribed in an ambulatory setting.

Throughout the discussion of the CARE level we have not mentioned diagnoses. Presently, we do not store them. Active development is proceeding at an international level on a diagnoses and complaint code for ambulatory care. We would prefer to wait until that solidifies before we commit ourselves to a large dictionary of diagnoses. Our plan is to present the active diagnoses as a list along the far right-hand column of bubbles in the bottom section of the encounter form.

Operational experience

The CIS has been in operation at the clinical level since July 1, 1973. Since then we have accumulated records on more than 2000 patients in four different clinics, the diabetes, the renal, one session of the general medicine, and the nurse clinician clinic, and have been providing the three previously mentioned reports for all patient visits to the above mentioned clinics. More than 200 different clinicians at varying professional levels—senior medical students, interns, residents, staff physicians and nurse clinicians—have been exposed to these reports in one or more clinics. These care providers have been cooperative and have complied to the rigid requirements for inscribing block-print numerics for direct optical reading into the computer. Both the initial and final sort is a computer rather than human sort. Multiple reports for multiple purposes (blood drawing lists, label, master logs, physician reports, ward reports, etc.) can be produced at will by specifying a command string. The system is flexible, robust and simple.

At the present, all tests performed by the chemistry, hematology and portions of serology section are handled by the computerized system. Test results for patients whose records are maintained by CIS are saved for batch updates to patient records.

The pharmacy module has been operational on a pilot basis since October 1974. A patient within the CIS presents his encounter form which bears his prescriptions to the pharmacy. After the necessary identifying information for the patient has been entered at the terminal, the computer cycles through all the active drug orders for the patient. These orders match the prescriptions printed on the request form (i.e., re-sorted by patient), and distributed to their originators. In our system, test requests are logged via a terminal. The worksheet is a turnaround document produced by the computer on which results are inscribed as block-print numerics for direct optical reading into the computer. Both the initial and final sort is a computer rather than human sort. Multiple reports for multiple purposes (blood drawing lists, label, master logs, physician reports, ward reports, etc.) can be produced at will by specifying a command string. The system is flexible, robust and simple.

Ancillary service modules

As was mentioned previously, the most efficient method of trapping data which flows from the ancillary services to the clinical level is to computerize the ancillary services. This is our goal. At present we have two such systems in operation.

The lab system has been reporting results to the ward since November of 1974. Information processing in the lab consists of a two cycle process. Test requests arrive at the lab grouped by patient. The requests must be re-organized to conform to the organization of the lab, that is, grouped by test station, each of which performs a single battery of tests. Common to every lab is the fact that an individual worksheet is produced for each work station. The worksheet is simply a list of patients whose samples are to be analyzed at a given work station. The analysis is done; the results are recorded on the worksheet, transferred to the request form (i.e., re-sorted by patient), and distributed to their originators. In our system, test requests are logged via a terminal. The worksheet is a turnaround document produced by the computer on which results are inscribed as block-print numerics for direct optical reading into the computer. Both the initial and final sort is a computer rather than human sort. Multiple reports for multiple purposes (blood drawing lists, label, master logs, physician reports, ward reports, etc.) can be produced at will by specifying a command string. The system is flexible, robust and simple.

The pharmacy module has been operational on a pilot basis since October 1974. A patient within the CIS presents his encounter form which bears his prescriptions to the pharmacy. After the necessary identifying information for the patient has been entered at the terminal, the computer cycles through all the active drug orders for the patient. These orders match the prescriptions printed on the encounter form, and the pharmacy clerk simply discontinues, repeats or changes the orders as the physician has indicated. New prescriptions are entered after all the active orders have been processed. For each prescription, the computer translates the prescription as written by the doctor (the sig.) into instructions to the patient, selects the tablet or bottle size from the pharmacy inventory which

From the collection of the Computer History Museum (www.computerhistory.org)
best fits the specified dosage, calculates the amount of medication to dispense, and prints a label for the medication bottle. Additional features aid the pharmacy in controlling its inventory.

**SYSTEM IMPLEMENTATION**

Since space does not permit a total description of the implementation of CIS, we will limit our description to some of the distinguishing features.

**Data base management system (DBMS)**

With the exception of the patient's medical record file and a few minor files, all data storage and retrieval is driven by a table of file descriptions and performed by a single I/O function and a set of encode/decode functions. The data base is divided into a number of logical files which may be contained in one or more physical files. All records contained in a single logical file are of the same size and format. A record may be between 2 and 512 bytes in length and can be divided into as many as 128 fields. Fields are of 9 types, scaled integer (2 bytes), floating point, variable sized scaled integer (from 1 to 7 bytes), date, time, pointer, string, packed string (only alpha-numerics) and bit string. Application programs access fields symbolically. Using the stored file descriptions, the I/O function determines the sector address, byte offset and length of the data requested. The data type is checked and the field is decoded if necessary. Records may be related in tree structured hierarchies (i.e. tree entry scheme). The DBMS maintains all pointers for the application programs. From any record tree, a program can access fields in that record or fields in any record in the subtree below it. The I/O function performs error checking and provides some security in that only authorized programs are allowed to change the contents of a given logical file. For direct addressing of records by content, a hash table is created. A logical file is required for each hashed field.

We have a set of utility programs for storing, editing, listing, sorting, rebuilding and packing the files maintained by the DBMS. A command language is used to specify the action to be performed, the path to be taken through the tree structure, and the fields to be accessed. The fields are specified by logical identifiers, and if we had the core space, could be represented by text names. Unfortunately, with our limitations numeric “names” must do. The application programmer does not enjoy the luxury of a command string. He must pass a variable length integer array to the I/O function.

The advantages of this DBMS are twofold. First, file maintenance is simplified since common utility programs are used for all of the different DBMS files. And second, program code is decoupled from file structure. The time overhead to the system is insignificant. We are I/O bound and the DBMS is efficient in its use of I/O. However, there is a significant core space overhead required by the I/O function and the core resident file descriptions.

**The statistical module**

Since tallies, counts and averages are important to the legislative process at every level, we have developed a statistical processor for our data bases. Using a command string analogous to that presented above, one chooses the fields of interest from his logical file. Three different statistical outputs can be produced for each of the specified fields: (1) averages, (2) standard deviations, and (3) histograms. Histograms are defined over a partition of either equal or unequal spacing. A set of cut points is entered to define the desired partition. Averages and standard deviations are only defined for numeric fields. The histogram output can be generated for any field type, including strings. Thus, we can create histograms of the age of our patient population tallied over ten year intervals, or that of the alphabetic distribution of names tallied over the letters of the alphabet.

Many of our statistical questions are of a conditional nature. For instance, one may not want the average blood pressure of the population, but the average blood pressure by patient age. The statistics module can produce such information. For conditional statistics, the user must specify conditional fields as well as target fields. A partition defined by cut points must be specified for each conditional field. By sorting the records of interest by the conditional fields, the task of generating conditional statistics is converted into a simple repeated application of the process defined above.

The statistical module can be used by all files in the CIS. For administration, it can provide work volume statistics by clinic, by parameter, by patient type, by time, etc. For the clinical lab, it can provide quality control statistics. For the clinician, it can provide drug usages statistics and the incidence of test abnormalities.

**Hardware and software**

The computer which supports the CIS is a Digital Equipment Corporation PDP 11/45 located in the Regenstein Institute at Marion County General Hospital. The PDP 11/45 is a midi computer with a word length of 16 bits and a cycle time of 300 nanoseconds. It operates as a time shared system available seven days a week, twenty-four hours a day except for periodic maintenance. Routine time sharing operations have been carried out on our system for more than two years. The central processor has 80K words of core memory, and 3.6 million words of disk memory. By the time this report is published, we will have acquired a 43 million word mass storage disk with expansion capability to 344 million words. The operating system, supplied by the vendor, is called RSTS/E. It provides a multi-user, interpretive environment for programming in BASIC-PLUS, which is a very powerful
extension of Dartmouth BASIC. It includes full string processing, flexible file handling, matrix instructions, Algol-like statements such as IF-THEN-ELSE, WHILE and UNTIL loops, and recursive subroutines. This language was designed as an application's language and has shown itself to be very suitable to our medical applications.

CONCLUSION

In summary, we are in the process of developing a CIS to serve all levels of ambulatory care, and the ancillary service and administration levels of inpatient care. In contrast to other such developments, we rely on computer printed and optically read turn-around documents rather than keyboard terminals for data input. We feel that a CIS can be much more than a passive repository of data, that it can be actively involved in executive decisions, particularly at the clinical level. The CARE language was developed for that reason. The use of a computer from a family of minis and midis provides great market flexibility for our system. At the lower end of the DEC PDP 11 line are systems which could be afforded by small group practices, and at the upper end, we believe, are systems which will be able to support large outpatient clinics such as our own. We are happy with our choice of application language. The overhead of an interpretive language is no drawback in an I/O bound system such as ours; the increased programming efficiency more than makes up for the short-comings of BASIC-PLUS.

We have not yet reached our goals. The data base must yet be extended to include diagnoses and narrative descriptions, the latter of which we will store in a tree structured code. Ancillary service modules are still to be developed for radiology, EKG, biopsy reports and nuclear medicine. Our patient population must be enlarged tenfold to encompass our clinical target population. We acknowledge that our goals are ambitious for a midi computer and an interpretive language. However, our experience to date suggests that they are within reach.

REFERENCES