An initial operational problem oriented medical record system—For storage, manipulation and retrieval of medical data*

by JAN R. SCHULTZ, STEPHEN V. CANTRILL and KEITH G. MORGAN

University of Vermont
Burlington, Vermont

INTRODUCTION

The ultimate role of the computer in the delivery of health services has yet to be defined. There may be profound implications in terms of quality of medical care, efficiency, economics of care, and medical research. Final judgments as to advisability and economic feasibility await the implementation of prototype total medical information systems and further technical developments directed toward lowering the high cost of currently developing systems. Development of less expensive hardware and real-time application of the present hardware and software must go on in parallel. We have been involved in the latter, and an experimental time-shared medical information system has been developed for storing and retrieving the total medical record, including both the narrative and the numeric data. This development has integrated the Problem Oriented Medical Record, a means of organizing medical data around a patient's problems, with a touch sensitive cathode ray tube terminal that allows structured input (with additional keyboard entry capability) by directly interfaced medical users (in particular the physician and the nurse).

A total of 85 general medical patient records have been kept on the system as of December, 1970. The system handles all aspects of medical record keeping—from the Past Medical History and Systems Review collected directly from the patient to complete Progress Notes and flow sheets, all recorded in a problem-oriented manner. It allows the direct inputting of data by the information originator and the retrieval of data in various medically relevant forms.

THE PHILOSOPHICAL BASIS FOR THE SYSTEM

The system is based on a medical philosophy requiring data in the medical record to be problem oriented and not source oriented.1 Data are collected, filed and identified in a problem oriented record with respect to a given problem and not to the source of the data as in the traditional source oriented record.

The problem oriented medical record requires a systematic approach to treatment of the patient. This systematic approach is defined by the four phases of medical action: data base collection, problem formulation, plan definition and follow-up. A brief explanation of the four phases will outline the basic requirements for this system (see Figure 1):

* This research was done under a research grant from the Department of Health, Education, and Welfare, Health Services and Mental Health Administration, National Center for Health Services Research and Development, PHS 1 R18 HS 00175-01, entitled "The Automation of a Problem Oriented Medical Record," Co-Principal Investigators Lawrence L. Weed, M. D. and Jan R. Schultz.
During the first phase of medical action the patient's complete data base is collected. This includes a branching questionnaire Past Medical History and Systems Review taken directly by the patient, a Physical Examination entered by the physician and other medical personnel, the Present Illness structured from choices (to be discussed later) and entered by the physician, and certain admission laboratory orders generated by the physician.

After a complete data base is collected the physician studies it and formulates a list of all the patient's problems. This is the second phase of medical action. The problem list includes medical, social, psychiatric and demographic problems. Each problem is defined at the level the physician understands it. A problem can be a "diagnostic entity," a physiologic finding, a physical finding, an abnormal laboratory finding, or a symptom. The problem list is a dynamic index to all the patient's plans and progress notes since it can be used to follow the course of the problem(s).

After a complete problem list is formulated, the physician must define an initial plan for each problem. This is the third phase of medical action. The plans are divided into plans for more information, plans for treatment and contingency plans. With plans for more information it is possible to: (1) rule out different problems by ordering certain tests or procedures, (2) get laboratory tests for management, and (3) get more data base information. Under plans for treatment: a drug, diet, activity, procedure, or patient education can be prescribed. Contingency plans are possible future plans to be carried out if certain contingencies are satisfied.

The fourth phase of medical action is writing progress notes for each problem. The progress notes for each problem are divided into: Symptomatic, Objective (laboratory, x-ray and other reports), Treatment Given, Assessment and Follow-up Plans (similar in content to the Initial Plans) sections. The progress notes allow medical personnel to act as a guidance system and follow the course of each problem, collecting more data base, reformulating and updating problems and respecifying the plans, each action dependent upon the course of the patient's problems.

The Problem Oriented Medical Record has been used in paper form for the past fourteen years. It has proved a working record system on paper and was demonstrated practical long before computerization was ever considered. Its dynamic structure, non-source orientation and medically relevant labeling of all data, however, facilitates computerization. Computerization augments its medical capabilities by making it possible to retrieve all data on one problem in sequence and by allowing data to be organized separate from its source in the record. This ability is recognized as having significant medical implications, for it allows the physician to follow the course of a problem in parallel with the patient's other problems or as a separate problem (i.e., retrieving information chronologically vs. retrieving all data on one problem). The computer enables rapid audit of all the patients with similar problems as well as the ability to audit a physician's logic and thoroughness on one specific patient, and will allow the development of research files.

As previously written:

"We should not assess a physician's effectiveness by the amount of time he spends with patients or the sophistication of his specialized techniques. Rather we should judge him on the completeness and accuracy of the data base he creates as he starts his work, the speed and economy with which he obtains patient data, the adequacy of his formulation of all the problems, the intelligence he demonstrates as he carefully treats and follows each problem, and the total quantity of acceptable care he is able to deliver."

Our experience with this system indicates that computerization does facilitate such an assessment.

At the time this project began, and the system was specified, other operating systems (of which we had knowledge) were few. After an analysis of these systems our group decided that we would try to build the necessary medical application programs using as a basis system software developed by another group. To quote R. W. Hamming;

"Indeed, one of my major complaints about the computer field is that whereas Newton could say, 'If I have seen a little further than others it is because I have stood on the shoulders of giants,' I am forced to say, 'Today we stand on each other's feet.' Perhaps the central problem we face in all of computer science is how we are to get to the situation where we build on top of the work of others rather than redoing so much of it in a trivially different way. Science is supposed to be cumulative, not almost endless duplication of the same kind of things."

This would serve two purposes: It would divide the total work task naturally into more manageable units, and it would force our group to learn completely the system we were to build upon, thus allowing the basic system software to be utilized as a tool in the accom-
plishment of our medical goals. We built upon the system being developed by Medical Systems Research Laboratory of Control Data Corporation. The hardware developments of Dr. Robert Masters (the Digiscribe) and the software developments of Mr. Harlan Fretheim (the Executive, the Human Interface Program and SETRAN) have been fundamental to our own progress.

THE BASIC SYSTEM SOFTWARE DEVELOPED BY MEDICAL SYSTEMS RESEARCH LABORATORY OF CONTROL DATA CORPORATION

Directly interfacing busy medical personnel to the computer system required the development of an effective facile interface. The Digiscribe terminal with its associated software is such an interface. Displayed on the cathode ray tube is an array of choices from which the user can make a selection by touching the screen with his finger. The user’s selection is input to the computer system in the form of a character so that for each of twenty different positions on the cathode ray tube screen a different character is generated. A system program accepts as input the user selection and on the basis of it appropriate branching takes place and new information is displayed. Included in the information used by the program that interprets the selection (the Human Interface Program) is a push-down list of frame numbers waiting to be displayed, branching information and certain internal parameters (not seen by the user at the terminal) which can be associated with each choice displayed on the screen. In addition to text, branching information and internal parameters, a program can be associated with any selection. This allows a certain amount of open endedness and provides the means for calling and executing application programs.

The selections made by the medical user at the terminal are concatenated by the Human Interface Program to form “paragraphs” of information. The paragraph is the basic unit of information generated in the system. (The Storage and Retrieval programs manipulate paragraphs of information.) Associated with each paragraph is the Selection Parameter List which includes for each choice made by the user the frame number, the choice number within that frame, and any internal parameters associated with the choice. The internal parameters can be used to code selections so programs can interpret compact codes rather than alphanumeric data. The internal parameters are identified by a single letter (e.g., “F” type internal parameters specify format codes which will be explained in detail below). Using the Selection Parameter List our programs can analyze the input data without having to search the generated English text.

The user generated paragraphs and associated Selection Parameter Lists, are the coupling mechanism between the Human Interface Program and our application programs which store, retrieve, and manipulate the patient records and other files.

A language was developed as part of the basic system called SETRAN (Selection Element TRANslator) which makes possible the programming of the branching logic displays and alteration of already entered displays using the keyboard on the terminal. See Figure 2 for an example of a frame as displayed in the Human Interface Program and in SETRAN. It has been possible to train physicians and other personnel in this language and development of new displays has proceeded without a computer person acting as an intermediary. Allowing medical personnel, with almost no computer science training a direct means of developing and altering basic system displays is fundamental during the development phase of systems such as these. The massive number of such displays required for such a system (currently over 16,000 displays have been developed by the PROMIS group) and the necessity for a tight feedback-loop during the developmental phase of the displays require such tools. Once systems such as this are beyond the developmental phase, access to such programs must be carefully controlled.

The operating system supports multiple terminals.
The "HUMAN INTERFACE PROGRAM (HIP)" displays medical content from the "FRAME DISPLAY DICTIONARY" to the user in a branching logic fashion. Most of the entries in the "FRAME DISPLAY DICTIONARY" were created by system development personnel at some time in the past by using the "SETTRAN PROGRAM."

As the user makes a series of choices at the terminal, "HIP" creates a "SELECTION PARAMETER LIST AND PARAGRAPH SEGMENTS" which represent the history of choices made by the user.

At appropriate times in the series of displays seen by the user, dependent upon pathway, "HIP" calls the "STORE PROGRAMS" and the "RETRIEVE PROGRAMS."

The "STORE PROGRAMS," by processing the "SELECTION PARAMETER LIST AND PARAGRAPH SEGMENTS" for the specific user, can update the mass storage resident "PROBLEM ORIENTED MEDICAL RECORD PATIENT STRUCTURED AND LIST FILES."

If the "STORE TRANSLATED" module of the "STORE PROGRAMS" is used (e.g., for processing patient histories), reference will be made to the "TRANSLATION DICTIONARY" which was created in the past by system development personnel using the "DETRAN PROGRAM."

The "RETRIEVE PROGRAM (STRUCTURED FILE)" processes the user's retrieval request as specified in his associated "SELECTION PARAMETER LIST AND PARAGRAPH SEGMENTS" retrieving any specified part of a patient's medical record.

Output by the "FORMAT ROUTINE" from the "RETRIEVAL PROGRAM (STRUCTURED FILE)" can be either to "PRINTED MEDICAL OUTPUT" (hardcopy) or directly to the user's cathode ray tube terminal via the "FRAME DISPLAY DICTIONARY" and "HIP."

The "RETRIEVE PROGRAM (LIST FILE)" may be called in the process of the user making choices at the terminal. It produces output directly to the user's terminal.

All of the above processes take place unknown to the user but are a direct response to his choices at the terminal in either storing in or retrieving information from the Computerized Problem Oriented Medical Record.

A rapid response to user interaction (a user familiar with the frame content can make selections faster than one per second) and supports application programs operating in a multi-level, multi-programming mode.

The frames, application programs, medical files and station associated variables are disk resident. Most selections made by a user require four disk accesses (in the current version of the system) and the variables associated with a station are core resident only while the selection for that station is being processed.

There are four main classes of application programs. The two highest level classes are interactive with the user and require immediate execution. The Selectible Element Translator is of this type and allows the online entering and changing of frame content and branching (by appropriate personnel, not all users).

The second level class of programs is executed while the user is at the terminal, but these programs may take longer to run than the ones executed immediately. An example of this type is the program which retrieves from the patient records to the cathode ray tube terminal.

The lowest level of application programs is executed sequentially by priority level in the background after a user has signed off the terminal. These include the programs which store into the patient records and retrieve patient records to the line printer.

AN APPROACH TO THE COMPUTERIZATION OF THE MEDICAL RECORD

Our approach to the computerization of the medical record involves many elements. The Problem Oriented Medical Record represents the medical "foundation" upon which the total system rests. The Human Interface Program, the Selection Element TRANslator, the system executive and the hardware "drivers" represent the computer software "foundation." The other elements to our approach will be discussed in this section.

The information generated by having the medical personnel (or the patient himself) go through the branching displays is English narrative. Although the number of selections presented on each display is small (averaging 8 selections) the number of paths through
these selections is quite large. There are currently over 16,000 displays in the system. Approximately 12,000 of these are branching displays, the remainder are solely for information, e.g., before any drug can be ordered a sequence of displays requests the physician to: (1) CHECK THE PROBLEM LIST FOR: followed by a list of problems (2) SIDE EFFECTS TO WATCH FOR: (3) DRUG AND TEST INTERACTIONS: (4) TEST INTERFERENCE: (5) USUAL DOSAGE: followed by optional (6) MECHANISM OF ACTION: (7) METABOLISM AND EXCRETION.

Structured branching logic displays allow the medical user to operate from a body of knowledge broader than can be kept in his own memory. This body of knowledge as represented in the library of displays is capable of being updated in an organized and systematic way, so that it can always reflect the most current and sophisticated medical thinking. The system is dynamic and since data can be typed into individual patient's records supplementing the structured displays, it would be possible to analyze this typed in data to update the library of displays (if the typed in data indicate a deficiency in the branching displays and not in the physician using the displays). Once systems similar to the one described here are in daily operation, the organized and systematic updating of the library of displays will have to be centralized in an organization with this sole responsibility and authority.

The generation of English is the result of a user making selections from structured, branching-logic displays. These selections must then be transformed by a program into an internal form for storage in a patient's file. The program that does the transformation of the selections into an internal form is generalized; it is independent of the specific content contained in the selections. It stores data in an internal form, independent of the output devices ultimately used to display the data. The retrieval routines that allow the stored data to be manipulated and displayed in various forms will also be discussed in more detail. (See Figure 3).

The displays necessary for the generation of each section of the Problem Oriented Medical Record are specified by a "meta-structure" for each section. The "meta-structure" specifies the branching logic of the content displays. There are structured approaches to Present Illness, Problem Lists, Drug Sequences, and Progress Notes. For example, the Present Illness metastructure would include, for each body system, a list of the symptoms particular to that system and for each symptom, a list of its characteristics. In the Psychiatry system, for example, if Headache were selected from the frame of the list of possible symptoms, the physician would be asked to describe characteristics of this headache:

If "MADE WORSE BY" were chosen by the physician, a frame containing the following selections appears:

This technique allows a complete English narrative description of HEADACHE to be generated:

1. Initial Operational Problem Oriented Medical Record System 243
All programs that interpret input data can assume a standard form and structure guaranteed by how the branching logic displays are programmed. Contained in the Selection Parameter List for each paragraph are the internal parameters which define the type of data and what the program should do with it. The programs also receive information which further describes the paragraph as a unit. This information was defined by selections as the user went through the displays. Associated with each paragraph is a paragraph label which further describes the paragraph on two levels: Information Type 1 (IT1) defines the major section in a record to which this paragraph belongs (e.g., Physical Examination, Progress Notes, Problem List, Past Medical History and Systems Review, etc.); Information Type 2 (IT2) defines the subsection within the major section (e.g., Skin Examination in the Physical Examination, Symptomatically in the Progress Notes, etc.). All paragraphs which contain problem-oriented data have associated with the label the applicable problem number. Also associated with the paragraph label is information indicating whether this paragraph contains either narrative or numeric information. The date, time, user and patient numbers are also associated. This data determines where the paragraph is to be stored in a specific patient's record.

Structured displays and the internal parameters linked to selections (and collected in the Selection Parameter List) both imply a closed system. The closed system enforces the organized entry of information in a well-defined syntax. Via the structured displays, the user is aware of data relationships that normally are imbedded in the data interpreting programs. The data is so entered that it has an inherent structure—not to be found when data is entered free form. This structure holds even if the data entered via selections from the structured displays are supplemented by typed-in information. Much simpler data interpreting programs are required for such structured data with the associated Selection Parameter List then for purely free formed input.

In addition, the user of structured displays can operate on recognition rather than recall. He has available, at the time he needs it, the organized knowledge of his profession. This knowledge can be systematically updated with a thoroughness that is impossible on an individual basis.

In summary, the necessary elements in our approach to a computerized system to store and retrieve Problem Oriented Medical Records include a medically relevant organization of the data in the medical record, an effective interface between the medical user and the computer system, a means of structuring the medical content material on frames using meta-structures, programs to transform selections into a manipulatable internal form, programs to retrieve the stored data in various forms, and a “closed” system.

FILE STRUCTURES FOR MEDICAL RECORDS

Our files may be characterized in terms of (1) maintenance and (2) structure. In terms of maintenance: Will this file handle information that can be both inserted and deleted (purgable) or only inserted (non-purgable)? An example of a file into which information will only be inserted and never purged is the patient’s problem-oriented medical record. A list of all the patients on a ward is an example of a file that will be added to and deleted from on a regular basis as patients are admitted and discharged from the ward. In terms of structure: We define a file of homogeneous elements (a file which contains a specific subset of the IT1, IT2's) as a list file; e.g., the list of patients on a specific ward, or the list of problems on a specific patient, or the list of current drug/diet/activities for a specific patient. A file of heterogeneous elements (all IT1, IT2's) we call a structured file, for example, the patient’s Problem Oriented Medical Record.

There are then a total of four types of files that we envision possible within the system: (1) a list file that cannot be purged; files of this type will ultimately be used for research retrieval capabilities; (2) a list file that can be purged; (3) a structured file that cannot be purged, and (4) a structured file that can be purged and which may in the future be used for the most current progress note. This progress note could be purgable since it might be possible, for example, to condense many vital sign values to one value and a range. For the current implementation of the system it has been necessary to utilize list files that are purgable and structured files that cannot be purged. Further expansion and sophistication of the current system will require the other two types of files to be developed and utilized. The two file types currently used will be described in more detail.

NON-PURGABLE STRUCTURED FILES—THE PATIENT’S PROBLEM ORIENTED MEDICAL RECORD

An individual patient's file requires a structure that facilitates the storage and retrieval of its data while minimizing the number of mass storage accesses. This file (a non-purgable file with heterogeneous elements) consists of a “Table of Contents” and a variable num-
The Table of Contents contains a variable length list of Item Pointers (see Figure 4). Each Item Pointer includes Information Type 1, the storage date and time, and the Item number containing the paragraph (5). If the Information Type 1 is problem oriented then there is also an array of bits that represent the presence of that problem number in the Item. This feature allows the rapid sequential retrieval of all the data on one problem by indicating whether the Item contains paragraphs on the problem number.

The Item is the depository of the narrative (e.g., Present Illness) and numeric (e.g., Blood Pressure) paragraphs. It contains the paragraphs generated by the user at the terminal and transformed by the STORE program into the internal form. (A description of this internal form is given later in this paper.) For each paragraph in the Item there is a corresponding paragraph pointer also in the Item. This paragraph pointer contains Information Type 1 and 2, problem number, date, time, user number and the relative address of the paragraph in this Item.

To access any specific paragraph of information in an Item a search through the paragraph pointers, which are sorted and linked together, is required. No searching of actual narrative data is necessary for the retrieval of any narrative paragraph contained in an individual Item, as the paragraph pointers completely define the contents of each narrative paragraph at the level the medical user needs to differentiate the data.

This type of file structure allows the retrieval of specific paragraphs of data in a minimum number of mass storage accesses (two accesses), if we assume the Table of Contents does not overflow. Since the Table of Contents (core resident in one access) contains Item Pointers for each Information Type in the patient's file, searching it gives the Item number(s) of the requested data. The necessary Item(s) can then be brought into core in one access per Item.

INTERNAL REPRESENTATION OF NARRATIVE AND NUMERIC DATA WITHIN THE PATIENT'S PROBLEM-ORIENTED MEDICAL RECORD

Each paragraph generated by the user to be stored into a patient's record can contain either narrative or numeric information. The STORE program and appropriate subroutines are used to transform each paragraph into a standard internal form which consists of strings of eight bit characters. The paragraph, for narrative data, represents the smallest unit of information the physician or other medical personnel can ever request on an individual patient. For numeric data, the numeric block—which specifies one numeric value—is necessary, as the physician requires time ordered graphs of various physiologic parameters and thus the retrieval of individual numeric blocks is required. It is possible to store a variable number of numeric blocks in one paragraph. In both cases the data are accessible at the level the medical personnel are most accustomed to working with it. This internal representation allows the rearrangement of data for various retrieval requirements that are impossible with a manual paper record system.

If the paragraph contains narrative data, a STORE routine interprets "F" type internal parameters in the Selection Parameter List as format codes. These format codes are associated with certain selections. They are used to define the internal form in which the data will be stored. This internal form in turn defines the output format of the paragraph and specifies the relationship between the selection with the "F" internal parameter and those selections following. The selection with the "F" internal parameter is treated as a "title" and the selections until the next "F" internal parameter are "data" (See diagram below). Specifically, the format code defines the indentation level of the title (level 0 is the least indentation, level 3 is the greatest amount) and whether there are carriage returns before and/or after the title. With this information it is possible to output the narrative using an interpretive output subroutine called FORMAT. The format codes and the internal form are output device independent.

Since SETRAN (Selection Element TRANslator) allows internal parameters to be associated with any
choice, the individual who writes the frame can specify
or change the output format. Such flexibility is an im­
portant feature in a system like ours as it allows the
individual writing the frame content material to spec­
ify, at the same time, the output format of the in­
formation. Changes of output format are also greatly
facilitated as they only require a rewriting, using
SETRAN, of the “F” internal parameters on the frame.
(Any information previously generated from these
frames is not reformatted.)

Internally a paragraph of narrative data is of the
following form:

(Format Code) (Title Narrative) (Data Narrative)...
(Format Code) (Title Narrative) (Data Narrative)...
(Format Code) (Title Narrative) (Data Narrative)
(Terminating Format Code)

For example:

(CR, LEVEL 0, CR) (SOCIAL PROFILE:)
(   LEVEL 1  ) (ADULT FEMALE. AGE 69.)
(   LEVEL 1  ) (BORN IN VERMONT; RURAL AREA. LIVED IN AREA OF CURRENT RESIDENCE FOR > 39 YEARS.)
(   LEVEL 1  ) (LAST COMPLETED GRADE: JUNIOR COLLEGE. WOULD NOT LIKE FURTHER EDUCATION OR TRAINING.)
(   LEVEL 1  ) (MARRIED. DOES NOT LIVE W/ HUSBAND. LIVES ALONE. COOKS OWN MEALS. WIDOWED FOR > 1 YEAR.)
(   LEVEL 1  ) (NOT SATISFIED W/PRESENT LIVING CONDITIONS.)
(   LEVEL 1  ) (UNEMPLOYED FOR MORE THAN 2 YEARS. DOES NON-STRENUOUS LABOR. GETS DAILY EXERCISE. PRESENT HEALTH CONDITIONS INTERFERE W/WORK.)
(   LEVEL 1  ) (SUPPORTED MAINLY BY SELF.)
(   LEVEL 1  ) (DOES NOT DRINK ALCOHOL.)

(   LEVEL 1  ) (EATS 2 OR MORE MEALS/DAY; MEAT OR EGGS WITH 1 OR MORE OF THEM.)
(   LEVEL 2  ) (22 POS, 12 NEG, 0 DNK, 0 DNU.)

(TERMINATION CODE)

A printout on the line printer of this paragraph would result in the following:

SOCIAL PROFILE:
ADULT FEMALE. AGE 69.
BORN IN VERMONT RURAL AREA. LIVED IN AREA OF CURRENT RESIDENCE FOR > 39 YEARS.
LAST COMPLETED GRADE: JUNIOR COLLEGE. WOULD NOT LIKE FURTHER EDUCATION OR TRAINING.
MARRIED. DOES NOT LIVE W/ HUSBAND. LIVES ALONE. COOKS OWN MEALS. WIDOWED FOR > 1 YEAR.
NOT SATISFIED W/ PRESENT LIVING CONDITIONS.
UNEMPLOYED FOR MORE THAN 2 YEARS. DOES NON-STRENUOUS LABOR.
GETS DAILY EXERCISE.
PRESENT HEALTH CONDITIONS INTERFERE W/ WORK.
SUPPORTED MAINLY BY SELF.
DOES NOT DRINK ALCOHOL.
EATS 2 OR MORE MEALS/DAY; MEAT OR EGGS WITH 1 OR MORE OF THEM.
22 POS, 12 NEG, 0 DNK, 0 DNU.

On the cathode ray tube it appears as:

SOCIAL PROFILE:
ADULT FEMALE. AGE 69.
BORN IN VERMONT RURAL AREA. LIVED IN AREA OF CURRENT RESIDENCE FOR > 39 YEARS.
LAST COMPLETED GRADE: JUNIOR COLLEGE. WOULD NOT LIKE FURTHER EDUCATION OR TRAINING.
MARRIED. DOES NOT LIVE W/ HUSBAND. LIVES ALONE. COOKS OWN MEALS. WIDOWED FOR > 1 YEAR.
NOT SATISFIED W/ PRESENT LIVING CONDITIONS.
UNEMPLOYED FOR MORE THAN 2 YEARS. DOES NON-STRENUOUS LABOR.
GETS DAILY EXERCISE.
PRESENT HEALTH CONDITIONS INTERFERE W/ WORK.
SUPPORTED MAINLY BY SELF.
DOES NOT DRINK ALCOHOL.
EATS 2 OR MORE MEALS/DAY; MEAT OR EGGS WITH 1 OR MORE OF THEM.
22 POS, 12 NEG, 0 DNK, 0 DNU.

The title and data narrative are both of variable length.
If the paragraph contains numeric data, then the
STORE-NUMERIC routine interprets “N” type internal parameters in the Selection Parameter List as numeric codes. The numeric codes associate specific medical data with the internal structure of the numeric blocks. The numeric blocks each contain numeric values or objective text; this could represent a blood pressure, a clinical chemistry value, or any other type of objective information that must be manipulated internally in the system. The numeric codes are used to associate
a type code (i.e., a number that represents the type of data contained in the block and is the means of identifying all numeric data within the system), a time, a date, a title, a numeric value, and objective text with the numeric block structure. For example, a temperature numeric block could contain: type code = 30; time = 14:35; date = Feb. 23, 1970; title = TEMP; numeric value = 38; number descriptor = C. Medical personnel writing frames can define and change the “N” type internal parameters using SETRAN. (Previously stored numeric blocks are not affected.)

The overall philosophy upon which the system was built required enough flexibility to allow the medical user, after a minimal training period on the system, to change all system variables specifying the output format (and thus the internal form of the data) and those internal parameters that depend upon medical knowledge of various physiological parameters. Medical personnel associated with our project can directly develop many of the system content frames and can change the specification of how the data will be stored internally by changing internal parameters entered with the frame content material. Such changes require no modifications to our programs.

PURGABLE LIST FILES—PATIENT, WARD AND MESSAGE FILES

There are two different types of purgable list files—the message files and the intermediate selectable list files.

To facilitate the closed nature of the system it must be possible to present previously entered dynamic data to the user on displays for selection (e.g., a list of a patient’s current active problems). The intermediate selectable list files perform this function. This ability to display lists for selection was necessary for us to develop because the basic system software (Control Data supplied) allows only for the creation of static displays (via the Selection Element TRANslator) from the keyboard and no ability to dynamically display various lists for selection. Because of time requirements in the displaying of lists, the data to be displayed come from this “intermediate” file rather than from scanning the entire patient’s file each time a list is to be created for a user.

The files are directly accessible on the basis of either the patient’s system I.D. number (patient list files) or ward number (ward list files). For each patient there are two classes of file entries: patient problem list entries and patient order entries. For each ward, the ward file consists of entries of the patients currently on that ward with additional information such as the status of each of their problems.

Two examples follow: To enter data on a patient, the patient must be identified. This could be done by having the user key-in certain identification information which then would be scanned, verified and used to access the patient’s record. The identification is more easily achieved, if the user knows the ward on which the patient is staying, by allowing the user to select the desired patient from a dynamic list of patients on a given ward. The selection, constructed from information in the patient’s record (including his name, age, sex, and unit number) has associated with it internal parameters that define this patient’s system I.D. number. This allows the STORE or the retrieval programs to directly address the patient’s Table of Contents. Another example, which has been very convenient for the nurses in reporting the administration of a drug on a patient, is a list of the current drugs that the patient is receiving.

The “message” file is not a pure “selectable” file since its entries are not used as part of the displays in the system. Its contents are copies of all the additions to the patient order files. In the future these could be sorted and printed out at the proper location in the hospital (e.g., laboratory, pharmacy, x-ray). However, this processing of the message files has not currently been implemented.

THE STORE PROGRAM FOR THE PATIENT’S STRUCTURED FILE

The STORE program stores all narrative and numeric data into the patient’s problem oriented medical record (structured non-purgable file). It is executed after the user at the terminal has confirmed as valid all generated paragraphs. No data are stored until after this final verification procedure. The STORE program receives as input the paragraphs and their associated Selection Parameter Lists. A Paragraph Index is built for all the paragraphs input to the program. Each paragraph’s identifying data: Information Type 1 and 2, problem number, storage mode, date, time, user number, and patient number are put in a Paragraph Index Element. After the Paragraph Index is built, it is sorted by patient number, date, time, Information Type 1, problem number, Information Type 2. This represents the order that the paragraphs are stored in the patient’s file (for normal retrieval).

For each Paragraph Index Element the storage mode...
defines which of these STORE routines is executed:

*STORE-DIRECT* is executed if the paragraph contains the narrative to be stored in the record. (This is the narrative shown on the top of the display as selections are made.) *STORE-DIRECT* interprets the “F” internal parameters in the Selection Parameter List as format codes and combines them with the selections to form the narrative data in its internal form.

*STORE-NUMERIC* is executed if the paragraph contains numeric blocks to be stored into the Item. *STORE-NUMERIC* interprets the “N” parameters in the Selection Parameter List along with the narrative in the paragraph to build the numeric blocks in their internal form.

*STORE-TRANSLATED* is executed if the paragraph is the result of a questionnaire. The paragraph does not contain narrative selections but the Selection Parameter List contains a record of the selections made on each frame. These paragraphs are formed when “YES” or “NO” questions are answered and the response must be translated into English narrative. The Selection Parameter List is interpreted and an “S” internal parameter associated with any selection signals a dictionary look-up using the frame and the choice number to define the dictionary element. The dictionary elements are concatenated according to rules defined in the dictionary and the resultant data are stored in the internal form for narrative data. Used in conjunction with *STORE-TRANSLATED* is a programming language similar to the Selection Element TRANslator. This program, the Dictionary Element TRANslator, DETRAN, is used to define the narrative to be associated with any choice, the rules to specify the concatenation of titles with subsequent dictionary elements, and the format codes necessary to specify the output format. Using *STORE-TRANSLATED* and Dictionary Element TRANslator it has been possible to give a patient a questionnaire in Spanish and have the narrative output in English.

THE STORE PROGRAM FOR THE LIST FILES

The STORE LIST program checks all newly input paragraphs to the patient’s record and determines if they should be used in updating the various intermediate selectable list files for that patient, the ward which he is currently on, or the message file.

This program (in its usual mode) takes as input, the paragraphs just stored into a patient’s record. This includes both narrative (e.g., problem statement) and numeric (e.g., order) paragraphs. From information in these paragraphs, the program may add, delete or alter entries in the appropriate (patient, ward, and message) list files. For example, if a new order is written for a patient, that order’s “text” along with the order problem number, the type code, the frequency and the number of times for administration, will comprise an entry which will be added to that patient’s intermediate selectable list file. The entries added to the file are sorted, using one or more elements of the entry as sort keys, depending upon the entry type (e.g., problem list entry, order entry, ward entry, etc.).

Although this program is usually called by the STORE program, it can be called independently of the STORE program, too. For example, a single patient’s intermediate selectable list files can be rebuilt by completely scanning all entries in the patient’s record.

RETRIEVAL PROGRAMS—RETRIEVAL OF DATA FROM THE PATIENT’S STRUCTURED FILE

The retrieval programs working on structured files are of two types: The first type retrieves both narrative and numeric data in the form of a narrative report; the second forms a time ordered “flowsheet” of various physiological parameters, clinical chemistry results and drugs administered. Both are strictly for the retrieval of data on a single patient.

Each retrieval program can display the retrieved information on either the cathode ray tube terminal or the high speed printer. Input to the retrieval programs is a paragraph which represents the retrieval request, that is, the complete specification of the data to be retrieved. Included with this paragraph is a string of internal parameters in the Selection Parameter List. The user is not aware of these parameters, he need only select the patient and the sections of the record desired (for example, Progress Notes on all active problems, History and Systems Review, or complete record grouped by major sections, i.e., at IT1 level). The retrieval program interprets the parameter string (which is well formed due to the structure of the retrieval frames).

Because each user must identify himself when he signs on, it is possible to allow him access to only certain displays in the system. Using this approach it is possible to limit an individual’s access to information within the system by allowing him to formulate only certain retrieval requests.

A retrieval may require one or more retrieval cycles depending on the number of major record sections (IT1’s) included in the request and the degree of grouping required in each major section. For each retrieval cycle required, the retrieval routine scans the Item pointers in the patient’s Table of Contents to determine which Items contain paragraphs satisfying this retrieval cycle. The Items are then brought into core in the order specified by the applicable Item pointers in the Table of Contents. For each Item the paragraph
 pointers are scanned, and for each paragraph pointer satisfying the current retrieval cycle request, the FORMAT routine is called to output the paragraph. The address of this paragraph is given to the FORMAT routine along with certain control information requested by FORMAT. The FORMAT routine interprets the paragraph looking for format codes and outputs it, continuing until terminated by the FORMAT termination code, then returning to the retrieval program. Once control is returned from FORMAT, the retrieval routine searches the Item list for the next proper paragraph pointer and continues feeding FORMAT until the list of paragraph pointers is exhausted. The retrieval program returns to the Item list, continuing until the Item list is exhausted.

A flowsheet is a time ordered table of multiple medical parameters. Sound interpretation of data involving clinical findings, vital signs, laboratory values, medications, and intakes and outputs requires organization of the data to clearly reveal temporal relationships and clarify the inter-relationships of crucial data. A user requests a flowsheet by selecting the patient, the medical parameters to be included on the flowsheet and the output device (printer or cathode ray tube). (See the flowsheet included in the annotated record.)

RETRIEVAL OF DATA FROM THE INTERMEDIATE SELECTABLE LIST FILES

Although technically a retrieval from the Intermediate Selectable List Files, the creation and presentation of Selectable Lists for the user is done in the context of his storing (or retrieving) other information to (or from) the patient’s record. For example, to write a Progress Note about a specific problem on the patient’s Problem List, the user must specify on which problem he is entering data. This is done by showing him, in display form, the list of the patient’s Current Problems and having him select the proper one. It should be noted that all information in the Selectable List was previously generated and stored by a user.

Input to this program includes the number (type) of Selectable List the user is to see. This number points to an entry in a table which then drives the creation of the frames in the display dictionary containing proper contents from the appropriate Intermediate Selectable List File. The user is then automatically shown the first display containing the list of elements. If more than one display is necessary, the additional displays are linked to the first display. The selection of the appropriate element in the list is then made under the Human Interface Program.

The complex of the Intermediate Selectable List Files with Store List and the subsequent creation of Selectable Lists allows information previously entered into the system to govern the storage and retrieval of other patient information, facilitating a closed system.

AN ANNOTATED EXAMPLE OF AN ACTUAL RECORD GENERATED ON THE SYSTEM

The following is an actual record from one of the patients on the computerized ward. This is a complete “cycled” record; i.e., data that have been added to a section are output chronologically within that section (e.g., page 6 includes the PMH & SR additions entered to the G.U./Renal and Neurology sections by the physician after reading the history). This printed output serves as the “paper” chart and is kept in the chart rack where the traditional paper record was kept. In this way a back-up record is always available, and attendings or consultants can utilize this paper record as well as the cathode ray tube terminal. This printed output is never written on and a new copy (or any updates) is printed daily.

The annotations associated with each page will help explain how the record is constructed, its relationship to the data as they are stored in the patient’s structured non-purgable file, and the user’s relationship to various aspects of the data as additional information is added to the record. In the annotations, the following abbreviations are used in specifying the different storage modes (SM):

SM = D Store Directly from selections or from keyboard.
SM = T Store Translated by a dictionary lookup based upon the frame number and the choice number.
SM = N Store Numeric from selections in an internal form which allows multiple numeric blocks within one paragraph (may include typed in information).

The purpose of duplication of the first page of the case is to show the layout and then the content on the same page. It would be helpful to refer to the Explanatory Legend for Figure 3 before proceeding. The blanked out spaces throughout the case are names which have been covered to protect the confidentiality of the patient.

ANNOTATED EXAMPLES OF THE SELECTABLE LIST FILES ASSOCIATED WITH THE PRECEDING RECORD

The following pages are copies as they appear on the cathode ray tube screen of the Selectable List on the same patient whose record has just been presented.
PATIENT PROFILE

NOTE: Because of the specified mode of retrieval ("cyclic"), each body system contains all HISTORY and LAB DATA BASE material entered under that body system in a cumulative manner, regardless of the time of entry as the aspect of the HISTORY and LAB DATA BASE (e.g., Lab order, patient HR). This ability of the Retrieve program allows an integrative association of subject-related information which is entered in a temporarily unrelated manner. This is facilitated by the STORE program storing the information in such a way that this and other associations (e.g., flow sheets output by INTERRVIEW-FLOWSHEET) are possible. This section of the record is "cycled" on each body system and may contain material entered HISTORY (SM=M) and/or physician entered LAB orders and reports (SM=R). This demonstrates that one body system under this ITI can contain all notes of storage (see GENTS-URINARY-RENS, below).

NOTE: This entire "PATIENT PROFILE" section was typed in by the physician using the "EDIT" (type-in) program. This section of the record is one of the forms which is not structured, allowed only typed-in information.

-- COMPUTERIZED POUR (ITER.2) -- SN 17/12/1970 10158 PAGE 2

PATIENT PROFILE

NOTE: This entire "PATIENT PROFILE" section was typed in by the physician using the "EDIT" (type-in) program. This section of the record is one of the forms which is not structured, allowed only typed-in information.

-- COMPUTERIZED POUR (ITER.2) -- SN 17/12/1970 10158 PAGE 3

CONSTITUTIONAL SUMMARY/GENERAL

NOTE: Because of the specified mode of retrieval ("cyclic"), each body system contains all HISTORY and LAB DATA BASE material entered under that body system in a cumulative manner, regardless of the time of entry as the aspect of the HISTORY and LAB DATA BASE (e.g., Lab order, patient HR). This ability of the Retrieve program allows an integrative association of subject-related information which is entered in a temporarily unrelated manner. This is facilitated by the STORE program storing the information in such a way that this and other associations (e.g., flow sheets output by INTERRVIEW-FLOWSHEET) are possible. This section of the record is "cycled" on each body system and may contain material entered HISTORY (SM=M) and/or physician entered LAB orders and reports (SM=R). This demonstrates that one body system under this ITI can contain all notes of storage (see GENTS-URINARY-RENS, below).

NOTE: This entire "PATIENT PROFILE" section was typed in by the physician using the "EDIT" (type-in) program. This section of the record is one of the forms which is not structured, allowed only typed-in information.

-- COMPUTERIZED POUR (ITER.2) -- SN 17/12/1970 10158 PAGE 4

NOTE: Because of the specified mode of retrieval ("cyclic"), each body system contains all HISTORY and LAB DATA BASE material entered under that body system in a cumulative manner, regardless of the time of entry as the aspect of the HISTORY and LAB DATA BASE (e.g., Lab order, patient HR). This ability of the Retrieve program allows an integrative association of subject-related information which is entered in a temporarily unrelated manner. This is facilitated by the STORE program storing the information in such a way that this and other associations (e.g., flow sheets output by INTERRVIEW-FLOWSHEET) are possible. This section of the record is "cycled" on each body system and may contain material entered HISTORY (SM=M) and/or physician entered LAB orders and reports (SM=R). This demonstrates that one body system under this ITI can contain all notes of storage (see GENTS-URINARY-RENS, below).

NOTE: This entire "PATIENT PROFILE" section was typed in by the physician using the "EDIT" (type-in) program. This section of the record is one of the forms which is not structured, allowed only typed-in information.

-- COMPUTERIZED POUR (ITER.2) -- SN 17/12/1970 10158 PAGE 5

NOTE: Because of the specified mode of retrieval ("cyclic"), each body system contains all HISTORY and LAB DATA BASE material entered under that body system in a cumulative manner, regardless of the time of entry as the aspect of the HISTORY and LAB DATA BASE (e.g., Lab order, patient HR). This ability of the Retrieve program allows an integrative association of subject-related information which is entered in a temporarily unrelated manner. This is facilitated by the STORE program storing the information in such a way that this and other associations (e.g., flow sheets output by INTERRVIEW-FLOWSHEET) are possible. This section of the record is "cycled" on each body system and may contain material entered HISTORY (SM=M) and/or physician entered LAB orders and reports (SM=R). This demonstrates that one body system under this ITI can contain all notes of storage (see GENTS-URINARY-RENS, below).

NOTE: This entire "PATIENT PROFILE" section was typed in by the physician using the "EDIT" (type-in) program. This section of the record is one of the forms which is not structured, allowed only typed-in information.

-- COMPUTERIZED POUR (ITER.2) -- SN 17/12/1970 10158 PAGE 6

NOTE: Because of the specified mode of retrieval ("cyclic"), each body system contains all HISTORY and LAB DATA BASE material entered under that body system in a cumulative manner, regardless of the time of entry as the aspect of the HISTORY and LAB DATA BASE (e.g., Lab order, patient HR). This ability of the Retrieve program allows an integrative association of subject-related information which is entered in a temporarily unrelated manner. This is facilitated by the STORE program storing the information in such a way that this and other associations (e.g., flow sheets output by INTERRVIEW-FLOWSHEET) are possible. This section of the record is "cycled" on each body system and may contain material entered HISTORY (SM=M) and/or physician entered LAB orders and reports (SM=R). This demonstrates that one body system under this ITI can contain all notes of storage (see GENTS-URINARY-RENS, below).

NOTE: This entire "PATIENT PROFILE" section was typed in by the physician using the "EDIT" (type-in) program. This section of the record is one of the forms which is not structured, allowed only typed-in information.
INJECTIONS IN PAST FIVE YEARS: INFLUENZA SMALL POX
IN PAST TEN YEARS HAS HAD POLIO SHOTS.
3 POS. 8 NEG. 0 DNU. 0 DNU.

FAMILY HISTORY/GENETIC:
- FATHER: STROKE- FATHER- HYPERTENSION- HYPERTENSION- MOTHER.
- FATHER DIED AT AGE 50 TO 59. MOTHER DIED AT AGE 60 TO 69.
- PATERNAL GRANDFATHER DIED AT AGE UNKNOWN. NOT KNOWN.
- MATERNAL GRANDFATHER DIED AT AGE 60 TO 69. NOT KNOWN.
- PATERNAL GRANDMOTHER DIED AT AGE NOT KNOWN. MOTHER GRANDMOTHER DIED AT
AGE NOT KNOWN.

DEMO-ALLERGIES:
- OFTEN WAVERED AROUND CHEMICALS, SOLVENTS, OR CLEANING FLUIDS.
1 POS. 4 NEG. 0 DNU. 0 DNU.

EYE: "MAY: NOSE, THROAT:
- WEARS OR HAS HORN GLASSES. NO GLASS STRENGTH SEEING UP CLOSE. VISION SATISFACTORY OF GLASSES.
3 POS. 17 NEG. 0 DNU. 0 DNU.
- MUCH LIT MUSE IN PLACE OF EMPLOYMENT. HAS HAD A GREAT DEAL
3 POS. 24 NEG. 0 DNU. 0 DNU.

DENTAL:
- MOUTH HEALTHY. EATS MUCH GREEN, POTATOES OR MACARONI. SOMETIMES EATS RAW
VEGETABLES. BRUSHES TEETH W/ TOOTHPASTE ONCE A DAY.
4 POS. 15 NEG. 8 DNU. 0 DNU.

H E A T H:
- C 0 R D I O V A S C U L A R: 0 DNU. 0 DNU. 0 DNU.
- G A S T R O I N T E S T I N A L: 0 DNU. 0 DNU. 0 DNU.
- E N D O C R I N E: 0 DNU. 0 DNU. 0 DNU.
- M U S C U L O S K E L E T A L: 0 DNU. 0 DNU. 0 DNU.
- S T O M A T O D: DENTAL: 0 DNU. 0 DNU. 0 DNU.

HISTORY:
- HAS TAKEN HYDROCHLORIC ACID, BORAX, SALT.
- HAS SHOT A GUN A GREAT DEAL.
- MUCH NOISE IN PLACE OF EMPLOYMENT.
- HAS TOOTHPASTE ONCE A DAY.
- HAS HAD POLIO SHOTS.
- INJECTIONS IN PAST FIVE YEARS: TYPHOID, PERTONAL STRAIN.

MEDICATIONS:
- HAS TAKEN HYDROCHLORIC ACID, BORAX, SALT.
- HAS HAD POLIO SHOTS.
- INJECTIONS IN PAST FIVE YEARS: TYPHOID, PERTONAL STRAIN.

NOTE: These are LAB orders under the RX & LAB DATA RANK (LR) section (DNU). These are the results stored as numeric blocks in the patient’s record by the STORE program. After each addition to a patient’s record, the STORE program calls STORE-LIST which then processes any entries to the patient’s record by using a REDEFINE routine. The new entries in this case, would be placed on a list of this patient’s outstanding lab. orders (see the list on pages 4,5,44 for example).

NOTE: These are the reported values for the previous two lab. orders, also stored as numeric blocks by the STORE program (DNU). These orders are reported by calling up the list (done by the VITAL-LIST program) of outstanding lab. orders for this patient as constructed by STORE-LIST. The set of stored orders is identified by the order number and the time it was ordered. The value for reporting the value is shown to the user when he chooses the order to report. This set of displays is pre-defined for each order at the time it is ordered by means of "*" parameters which cause the placement of a specific value in a specific field in the numeric block for that order. In most cases, these results can also appear on flowsheets. This is one way that the patient’s record can be kept up to date. For example, after these results have been stored in the patient’s record by STORE, STORE-LIST will process these numeric blocks, removing the "CRI" and "SBD REF" entries from the list of outstanding orders, since unless otherwise specified, lab. orders are assumed to be ended only once.

--- COMPUTERIZED FORM (LH.2) --- SN 12/18/70 19:55 PAGE 5

--- COMPUTERIZED FORM (LH.2) --- SN 12/18/70 19:55 PAGE 5

NOTE: These are LAB orders under the RX & LAB DATA RANK (LR) section (DNU). These are the results stored as numeric blocks in the patient’s record by the STORE program. After each addition to a patient’s record, the STORE program calls STORE-LIST which then processes any entries to the patient’s record by using a REDEFINE routine. The new entries in this case, would be placed on a list of this patient’s outstanding lab. orders (see the list on pages 4,5,44 for example).

NOTE: These are the reported values for the previous two lab. orders, also stored as numeric blocks by the STORE program (DNU). These orders are reported by calling up the list (done by the VITAL-LIST program) of outstanding lab. orders for this patient as constructed by STORE-LIST. The set of stored orders is identified by the order number and the time it was ordered. The value for reporting the value is shown to the user when he chooses the order to report. This set of displays is pre-defined for each order at the time it is ordered by means of "*" parameters which cause the placement of a specific value in a specific field in the numeric block for that order. In most cases, these results can also appear on flowsheets. This is one way that the patient’s record can be kept up to date. For example, after these results have been stored in the patient’s record by STORE, STORE-LIST will process these numeric blocks, removing the "CRI" and "SBD REF" entries from the list of outstanding orders, since unless otherwise specified, lab. orders are assumed to be ended only once.

--- COMPUTERIZED FORM (LH.2) --- SN 12/18/70 19:55 PAGE 5

NOTE: These are LAB orders under the RX & LAB DATA RANK (LR) section (DNU). These are the results stored as numeric blocks in the patient’s record by the STORE program. After each addition to a patient’s record, the STORE program calls STORE-LIST which then processes any entries to the patient’s record by using a REDEFINE routine. The new entries in this case, would be placed on a list of this patient’s outstanding lab. orders (see the list on pages 4,5,44 for example).

NOTE: These are the reported values for the previous two lab. orders, also stored as numeric blocks by the STORE program (DNU). These orders are reported by calling up the list (done by the VITAL-LIST program) of outstanding lab. orders for this patient as constructed by STORE-LIST. The set of stored orders is identified by the order number and the time it was ordered. The value for reporting the value is shown to the user when he chooses the order to report. This set of displays is pre-defined for each order at the time it is ordered by means of "*" parameters which cause the placement of a specific value in a specific field in the numeric block for that order. In most cases, these results can also appear on flowsheets. This is one way that the patient’s record can be kept up to date. For example, after these results have been stored in the patient’s record by STORE, STORE-LIST will process these numeric blocks, removing the "CRI" and "SBD REF" entries from the list of outstanding orders, since unless otherwise specified, lab. orders are assumed to be ended only once.

--- COMPUTERIZED FORM (LH.2) --- SN 12/18/70 19:55 PAGE 5

NOTE: These are LAB orders under the RX & LAB DATA RANK (LR) section (DNU). These are the results stored as numeric blocks in the patient’s record by the STORE program. After each addition to a patient’s record, the STORE program calls STORE-LIST which then processes any entries to the patient’s record by using a REDEFINE routine. The new entries in this case, would be placed on a list of this patient’s outstanding lab. orders (see the list on pages 4,5,44 for example).

NOTE: These are the reported values for the previous two lab. orders, also stored as numeric blocks by the STORE program (DNU). These orders are reported by calling up the list (done by the VITAL-LIST program) of outstanding lab. orders for this patient as constructed by STORE-LIST. The set of stored orders is identified by the order number and the time it was ordered. The value for reporting the value is shown to the user when he chooses the order to report. This set of displays is pre-defined for each order at the time it is ordered by means of "*" parameters which cause the placement of a specific value in a specific field in the numeric block for that order. In most cases, these results can also appear on flowsheets. This is one way that the patient’s record can be kept up to date. For example, after these results have been stored in the patient’s record by STORE, STORE-LIST will process these numeric blocks, removing the "CRI" and "SBD REF" entries from the list of outstanding orders, since unless otherwise specified, lab. orders are assumed to be ended only once.

--- COMPUTERIZED FORM (LH.2) --- SN 12/18/70 19:55 PAGE 5

NOTE: These are LAB orders under the RX & LAB DATA RANK (LR) section (DNU). These are the results stored as numeric blocks in the patient’s record by the STORE program. After each addition to a patient’s record, the STORE program calls STORE-LIST which then processes any entries to the patient’s record by using a REDEFINE routine. The new entries in this case, would be placed on a list of this patient’s outstanding lab. orders (see the list on pages 4,5,44 for example).

NOTE: These are the reported values for the previous two lab. orders, also stored as numeric blocks by the STORE program (DNU). These orders are reported by calling up the list (done by the VITAL-LIST program) of outstanding lab. orders for this patient as constructed by STORE-LIST. The set of stored orders is identified by the order number and the time it was ordered. The value for reporting the value is shown to the user when he chooses the order to report. This set of displays is pre-defined for each order at the time it is ordered by means of "*" parameters which cause the placement of a specific value in a specific field in the numeric block for that order. In most cases, these results can also appear on flowsheets. This is one way that the patient’s record can be kept up to date. For example, after these results have been stored in the patient’s record by STORE, STORE-LIST will process these numeric blocks, removing the "CRI" and "SBD REF" entries from the list of outstanding orders, since unless otherwise specified, lab. orders are assumed to be ended only once.
INITIAL OPERATIONAL PROBLEM ORIENTED MEDICAL RECORD SYSTEM 253

C02
CL

3.5 MEQ/L CO2
93 MEQ/L CL

112 MG/100 M L 2 HR. P.C. GLUC NOT STATED ON LAB SLIP WHETHER 2 HR PC

GENDO-URINARY/RENAI:

URINE HAS BEEN BLOODY OR COFFEE COLORED.
TOLD HAD KIDNEY OR BLADDER STONES. STONES PASSED WITH URINE.
TOLD HAD KIDNEY OR BLADDER INFECTION DOES NOT HAVE ANY SEX PROBLEMS.
HAS NOT BEEN CIRCUMCISED.

6 POS. 18 NEG. 0 DNK. 0 DNU.

TROUBLE MOVING ARMS AND LEGS ON BOTH SIDES.
NUMBNESS OR TINGLING OF ARMS, HANDS, LEGS OR FEET PRESENT FOR SEVERAL WEEKS.
5 POS. 18 NEG. 0 DNK. 0 DNU.

POSSIBILITY MOVING THE LEFT LEG IS BECAUSE OF THE PAIN ASSOCIATED WITH THE ARTHRITIS IN THE LEFT KNEE. DENIES NUMBNESS OR TINGLING IN THE ARMS AND LEGS.

PSYCHIATRY:

DECREASED INTEREST OR ENJOYMENT IN SEX.
2 POS. 34 NEG. 0 DNK. 0 DNU.

VITAL SIGNS:

TEMP. ORAL DEGREES C 36.8
PULSE, RADIAL: 88/MIN. REGULAR
RESPIRATION: 20/MIN.
BP. LT ARM, SITTING 170/105 MM HG.
BP. RT ARM. STANDING 160/105 MM HG.
JVP 8 CM ABOVE STERNAL ANGLE AT 45 DEGREES ELEV. WEIGHT: NOT DETERMINED. HEIGHT/LENGTH: NOT MEAS.

GENERAL APPEARANCE:
THE PATIENT IS A CHRONICALLY ILL, NORMALLY NOURISHED, MIDDLE AGE: (APPEARS STATED AGE). CAUCASIAN MALE, RESPONSIVE AND CAN CARE FOR SELF.
CURRENTLY REQUIRES IN D AIDS. PT IS ANXIOUS.

SKIN NORM.
HEAD NORM.
EYES:
EYE MOVEMENTS: EYE MOVEMENTS NORMAL.
PUPILS: ROUND, EQUAL, 4 MM, ESTIMATED. REACT TO LIGHT & ACCOM.,
FUNDUS: BILAT. NORM.
EARS:
EXTERNAL CANAL: BILAT. NORM.
TYPANIC MEMBRANE: BILAT. NORMAL COLOR, MID POSITION, LIGHT REFLS. NORM.
HEARING: NORMAL BILAT.
NOSE & NASOPHARYNX: NORM.
OROPHARYNX:
TONGUE: BILAT. ROUGH TRENCH, GENERALIZED DEEP TEE.
GENERAL: MANY ABSENT.

NOTE: These are the totals for the patient administered HISTORY sections as stored in the patient's record by the STORE program.

NOTE: These vital signs were entered as soon as the patient arrived on the ward.

NOTE: These vital signs and the rest of the physical exam were entered by the physician following the normal patient work-up on the ward.

SUMMARY STATISTICS:
INTERVIEW COMPLETED.
105 POS. 27 NEG. 0 DNK. 0 DNU.

PHYSICAL DATA BASE BY SYSTEM.

NOTE: These are the totals for the patient administered HISTORY sections as stored in the patient's record by the STORE program.

From the collection of the Computer History Museum (www.computerhistory.org)
NECK: THYROID NORMAL SIZED, SYMMETRICAL.
LYMPH NODES: NONE PAINFUL.
CHEST AND LUNGS:
RESPIRATION NORMAL.
INSPECTION NORMAL.
PULSATION NORMAL.
PERCUSSION: RESONANT THROUGHOUT.
AUSCULTATION:
NORMAL BREATH SOUNDS: BILAT. ENTIRE CHEST.
CARDIOVASCULAR:
ARTERIAL PULSES: ALL NORMAL.
JUGULAR VENOUS PULSE: VENOUS PRESSURE: 0 CM ABOVE STERNAL ANGLE.
PALPATION:
APEX BEAT LOCALIZED AT MCL IN 4TH ICS.
AUSCULTATION:
NORMAL.
ABDOMEN:
INSPECTION NORMAL.
PULSATION NORMAL.
AUSCULTATION NORMAL.
RECTAL:
STOOL ABSENT.
EXTREMITIES:
JOINTS:
WRIST: RT: SWELLING/ENLARGEMENT: OVER ENTIRE JOINT, MAINLY SOFT TISSUE, RUBBERY, TENDER, HOT.
JOINTS:
KNEE: LT: PAIN ON MOTION: ACTIVE & PASSIVE, CONSTANT, INFLAMMATION W/O SWELL.
JOINTS:
LOWER LEGS: Pitting Edema, Ankles, BILAT.
BACK/FLANK/SPINE:
PALPATION/PERCUSSION:
BILAT. MILD.
BILAT. SOFT TISSUE.
MALE GENITALIA:
PUBIC HAIR:
FEMALE ESCUTCHEON:
NEUROLOGIC EXAM:
MENTAL STATUS: FULLY RESPONSIVE, DAY OF WEEK, DATE OF MONTH, FULLY COOPERATIVE.
ACTIVITY & BEHAVIOR: NORMAL MOTOR ACTIVITY, ABLE TO CARE FOR SELF, NORMAL BEHAVIOR.
APPEARANCE: UNORIENTED, SLOPPY.
MENTAL STATUS: FULLY RESPONSIVE.
HALLUCINATIONS: ABSENT.
Mental Status: Full.
INTELLIGENCE:
ABLE TO ABSTRACT.
FUND OF KNOWLEDGE: Slightly Deficient.
SPEECH: NORMAL.
CALCULATIONS: DOES SERIAL 7'S.
INSIGHT & AWARENESS: UNDERSTANDS ILLNESS, JUDGEMENT NORMAL, MOTION EFFECTIVE, WITHDRAWN, FLAT.
THOUGHT CONTENTS: NORMAL.
OWNERSHIP: NORMAL.
II: NOT TESTED.
III: NOT TESTED.
IV: COLOR VISION NOT TESTED.
V: VISUAL FIELDS NORMAL.
VI: HEARING NORMAL.
VII: TASTE NOT TESTED.
VIII: HEARING NORMAL.
IX: Hearing Normal.
X: Hearing Normal.
XII: TREMOR, BILAT.
MOTOR:
STRENGTH: NORMAL.
BULK: ATROPIV, SLIGHTLY BILAT.
COORDINATION:
NORMAL.
GAIT, STANCE: NORMAL.
ABNORMAL MOVEMENTS: TREMOR, ENTIRE BODY, AT REST.
REFLEXES:
BICEPS JERK: BILAT.
TRICEPS JERK: BILAT.
KNEE JERK: BILAT.
ANKLE JERK: BILAT.
PLANTAR RESPONSE: RT FLEXOR, LT FLEXOR.
ABNORMAL:
SENSORY:
COOPERATION GOOD.
SUPERFICIAL PAIN:
NORMAL.
TOUCH:
NORMAL.
VIBRATION:
MODERATELY DECREASED RT SIDE, LT SIDE.
VIBRATION:
SLIGHTLY DECREASED LT SIDE.

---

<GEN WARD INFO>
PLACEMENT:
UP BY ASSISTANCE 4AC 10B
VISITING REGULAR
MOBS 14/ 19/28 11/19/70
NOTE: "GENERAL WARD INFORMATION" contains information necessary for general ward/level functions. This section is not problem oriented.
From the collection of the Computer History Museum (www.computerhistory.org)
Initial Operational Problem Oriented Medical Record System -

**Pulse OIH**  
**Resp OIH**  
**BP OIH**  
**JU**

**CHLORAL HYDRATE 1000**

37.4°C TEMP OIH  
88/MIN PULSE OIH  
20/MIN RESP OIH

142/100 BP OIH TAKEN AT 8100 AM

10/19/70

**NOTE:** This is the execution of the "CHLORAL HYDRATE" ordered immediately above. This execution was specified by the Unit Nurse (UN025) by choosing the order from the outstanding list of orders displayed by the ✮VISUAL LIST program from the list created by STORE-LIST. Note that the entire order is not shown, but rather is truncated with ✮. This was done to make the record easier to read; the entire order is always available in the record. The order is assumed to be executed as stated unless otherwise stated (e.g., blood pressure (BP) entered by UN025 at 10:58). Since this order was specified to be executed only once ("X1"), the order is removed from the list of outstanding orders when the numeric block reflecting the execution (stored by ✮STORE-LIST) is processed by ✮STORE-LIST.

37.4°C TEMP OIH  
88/MIN PULSE OIH  
20/MIN RESP OIH

160/90 BP OIH

88/MIN PULSE OIH  
20/MIN RESP OIH

138/90 BP OIH  
88/MIN PULSE OIH  
20/MIN RESP OIH

164/90 BP OIH  
42/MIN PULSE OIH  
22/MIN RESP OIH

**CHLORAL HYDRATE 1000**

37.2°C TEMP OIH  
88/MIN PULSE OIH  
20/MIN RESP OIH

160/90 BP OIH

**<DONE> PT.ACT: UP W/ ASSISTA**

37.4°C TEMP OIH  
88/MIN PULSE OIH  
20/MIN RESP OIH

92/MIN PULSE OIH  
2/MIN RESP OIH  
180/100 BP OIH

7135 11/23/70

37.4°C TEMP OIH  
88/MIN PULSE OIH  
20/MIN RESP OIH

7157 11/24/70

37.4°C TEMP OIH  
88/MIN PULSE OIH  
20/MIN RESP OIH

76/MIN PULSE OIH  
2/MIN RESP OIH  
166/104 BP OIH

23110

104/MIN PULSE OIH  
20/MIN RESP OIH  
125/90 BP OIH TAKEN AT 8100 PM

88/MIN PULSE OIH  
178/100 BP OIH  
20/MIN RESP OIH

36.9°C TEMP OIH  
82/MIN PULSE OIH  
28/MIN RESP OIH  
84/MIN PULSE OIH

21117

From the collection of the Computer History Museum (www.computerhistory.org)
From the collection of the Computer History Museum (www.computerhistory.org)
NOTE: Here is another order in the form of a request for a SOCIAL SERVICE CONSULT.

These are a few duplicate entries and errors in using the type-in program (KEY).

NOTE: This is the reply to the request for a SOCIAL SERVICE CONSULT entered by the consultant.

Additional information from consultant.

From the collection of the Computer History Museum (www.computerhistory.org)
improving overall.

obj

has seen the patient and the psychometric tests have been administered. the preliminary results suggest that there is some intellectural impairment, perhaps more than could be explained merely by age. but that the defect is not severe enough that it seems to be in keeping with the friends and relatives concerns. full report to follow.

admit

feel that patient is stable enough for discharge. we have reached the point of h 2 b matching hospital benefits, will be followed by me. we are following the economic opportunities and at this time:

improving overall.

obj

have called back and feels that patient can be discharged. he states that he has questioned whether the patient might be displaying signs of early pick's disease but seems satisfied with our psychologic evaluation. he has asked that the patient be sent home on Prednisone, for he feels that when he gets home he will probably quite antimonial.

plan-pat ed

pt will be told: nature of the problem: probable course over

therapy prescribed. might home on phenobarbital is no.

cum reply


social service note: we were asked to take pt. home. mr. still expressed concern regarding future of patient. he feels that patient still needs more pt. for ethological evaluation. he has asked that the patient be sent home on prednisone. for he feels that when he gets home he will probably be quite antimonial.

discussion: patient is guarded for psycho-social. has pt. this time. he feels that pt. will return to old pattern.

2. (labeled) (chronosis)

plan:


notes

2. (labeled) (chronosis)

obj

notes

since the information in the patient's record (as stored by the store program) cannot be deleted, any mistakes in entering information must be corrected by entering and noting the corrected information. in this case, the alkaline phosphatase value was originally entered incorrectly.

notes: this status is reflected in the selectable list (see page 4) of the patient's problem in terms of being able to look at all problems or those that are getting worse. it is also possible to look at lists of patients on a ward with problems in a certain subspeciality, problems getting worse or problems in a certain subspeciality getting worse.

notes: another entry by the consultant.

notes: assessment of a problem is always typed in.

notes: another entry by the consultant.
DEMA OF ANKLES CLEARING

METABOLISM OF EDEMA SOMEWHAT UNCLEAR WITH SERUM PROTEIN OF 6.2 AND ESSENTIALLY NORM. CHEST FILM. LOW BUN. ESSENTIALLY NORM. LFT'S. IN VIEW OF RE # METABOLISM OF EDEMA SOMEWHAT UNCLEAR SINCE LFT'S. BUN. CHEST FILM ALL ESSENTIALLY NORM. SERUM PROTEIN OF 6.2 NORM. HOWEVER EDEMA CLEARING AT PRESENT. AND ALSO EVIDENCE OF ARTHRITIS L ANKLE. PROBABLY DUE TO SPRAINING.

PLAN - INITIAL

4. INFLAM. ARTICULAR DISI INVOLVING: WRIST/CARPALS RT, KNEE, LT. SECONDARY TO:

PLN-R/DI /MD014/ 191ZB 11/19/70

N/0 B&W.
LATX X/F.
N/0 H/D.
ASLT.

N/0 INQUAD.
URIC AC.

N/0 INFECTIOUS (Doubt in view of absence of fever)

N/0 RT/WRIST.
N/0 WRIST.

N/0 KNEE LT.

N/0 KNEE.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.

ST-DIS.
JOINT PAIN SUBSIDING.
JOINT SWELLING SUBSIDING.

CAN DETECT LITTLE DIFFERENCE IN TEMP BETWEEN WRISTS AND KNEES; SWELLING HAS SUBSIDED, AND PATIENT WALKS ALMOST WITHOUT LIMP.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.

NOTE: Initial plans should have been specified for this problem.
The first list is the patient's Problem List. This is used to define which problem a Progress Note is to be written on or to define a specific problem for the retrieval of all data in sequence on one problem. The physician or nurse selects the specific problem and then proceeds to enter the data. The last two lists (for therapy and non-therapy orders) are used to indicate when something has been given and to allow the entry of laboratory and x-ray results.

Associated with each patient's computer-based record are these three Intermediate Selectable List Files. They contain copies of information also contained in the record. (Compare the Problem List of the printed record with the one on the screen and the Plans for problem # 8 in the printed record with the General Ward and non-therapy order list in selectable form on the cathode ray tube.) The data are stored redundantly in the lists to facilitate speedy retrieval.
THE PATIENT'S GENERAL WARD AND NON-THERAPY ORDER LIST IN SELECTABLE FORM ON THE CATHODE RAY TUBE SCREEN

NOTE: The problem titles and specific orders have been truncated. In case of ambiguity, it is possible to see the complete problem statement and/or the complete orders by referring back to the printed record or by a specific retrieval to the cathode ray tube terminal.

THE PATIENT'S THERAPY ORDER LIST IN SELECTABLE FORM

NOTE: The problem titles and specific orders have been truncated yet a complete copy of everything is contained in the record in case of any ambiguities.

CONCLUSION

An experimental time-shared medical information system has been developed upon the organizing principle of the Problem Oriented Medical Record. The boundaries of the system are defined by the content of the patient's Problem Oriented Medical Record and all the information that is a natural extension of that document, e.g., the billing, the pharmacy, laboratory and X-ray data, the admitting office information, etc.
The principle focus of our effort has been to directly interface the patient, the physician, and the nurse with the computer in a manner that allows the input and retrieval of all medical information normally generated by them for the medical record. The subsequent computer interfaces of the same medical information via message switching with the business office for billing, the pharmacy, the laboratories, admitting office, etc., in a meaningful problem oriented fashion are planned but not yet implemented.

Now that the physician-nurse component has been effectively integrated into a computerized system, final assembly of all the components is possible. In this regard, as Jay Forrester found in management and engineering, we are finding in medicine through the Computerized Problem Oriented Record and its many extensions that the amplification and interactions among the components of the system may at times be more important than the components themselves. We are now prepared to develop a model whereby the ultimate role of the computer in the delivery of health services can be defined.

ACKNOWLEDGMENT

We would like to acknowledge the medical philosophical leadership and clinical expertise of Lawrence L. Weed, M. D.; the clinical experience and expertise of Laura B. Weed, M. D. for the massive effort of developing the bulk of the medical content material (in the branching structured logic displays) in the system; the development of the drug displays by George E. Nelson, M. D.; the early computer software specification efforts of Mrs. Lee Stein; the medical superstructure specifica-
tion efforts of Charles Burger, M. D.; and the efforts, as represented by the actual patient record, of the house staff and nurses at the Medical Center Hospital, Burlington, Vermont. Important contributions have also been made by many other people whom we wish to thank for their efforts. The authors would like to state explicitly that the computer software described in the paper would be but a skeleton without the medical content material contained in the branching displays.

REFERENCES

1 L L WEED
Medical records, medical education and patient care
Case Western Reserve University Press Cleveland Ohio 1969

2 L L WEED
Medical records that guide and teach

3 M D BOGDONOFF
Clinical science
Archives of Internal Medicine Volume 123 p 203 February 1969

4 R M GURFIELD J C CLAYTON JR
Analytic hospital planning: A pilot study of resource allocation using mathematical programming in a cardiac unit
RAND Corporation RM-5853-RC April 1969

5 MASSACHUSETTS GENERAL HOSPITAL
Memorandum nine
Hospital Computer Project (Status Report) February 1966

6 C M CAMPBELL
Akron speeds information system slowly
Modern Hospital Volume 104 p 118 April 1965

7 R W HAMMING
One man's view of computer science
Journal of the Association for Computing Machinery Volume 16 pp 3-12 January 1969

8 CONTROL DATA CORPORATION
SHORT operating system program rules reference manual
Publication Number 60259600 April 1968

9 CONTROL DATA CORPORATION
SHORT operating system basic formats reference manual
Publication Number 60259700 February 1969

10 CONTROL DATA CORPORATION
SETTRAN selectable element translator reference manual
Publication Number 60249400 May 1968

11 L L WEED
Technology is a link not a barrier for doctor and patient
Modern Hospital pp 80-83 February 1970

12 L L WEED et al
The problem oriented medical record—Computer aspects (A supplement to Medical Records, Medical Education and Patient Care)
Dempco Reproduction Service Cleveland Ohio 1969