A History of the PROMIS Technology: An Effective Human Interface

Jan R. Schultz

Jan Schultz has lived in Burlington, Vermont for the last 17 years. He was associated with the University of Vermont from 1969 until 1981 as a Research Associate in the College of Medicine. He coordinated computer related activities at PROMIS Laboratory during that period and conducted research in programming languages and database systems. In 1981 he left the University and started PROMIS Information Systems, Inc.

Jan was born in Chicago, Illinois, and went to the University of Illinois. He majored in Mathematics and Philosophy and has a BS degree in Mathematics. While working on his master's degree he helped develop a very innovative computer-based teaching machine called Socrates. Enjoying his computer work, he combined computers and mathematics, studied-automata theory, and began work on a Ph.D. at Case Western Reserve University. While at Case, he met Larry Weed and decided to devote his full energies to what became the PROMIS project.

Editor's Note

[Editor's Note: Jan's preliminary comments are included as a preface to his paper.]

Coming late in the program gives me the advantage of seeing what spaces in computing have already been covered. The dimensions of the space that we, in fact, worked in are very different from a lot of the things that we've seen and discussed earlier in the program. One of the things we felt very strongly about early on was keeping data in an electronic form. And so, unlike Xerox PARC, we didn't have the influence, or perhaps the pressure, of having to get things out on Xerox machines.

We're really looking not at automating offices, but at automating companies. In our case, the company was a hospital. But it was a large organization, and we were trying to build tools that could match the variety of the organization. The tool had to have enough complexity in it to be able to control the organization, so it had to be able to maintain the variety of the organization within it. This really defined a fairly complex sort of a computer system that had to be able to do that. Like Butler Lampson, we were interested in interaction rather than programming for the users of our system. In trying to come up with a title for the paper, the notion of an effective human interface was as far as I could go. To try to put a touch screen in it was just too limiting because it isn't just the touch screen that makes the system effective, it's a touch screen with a large database in a very interactive environment. It's a combination of all those three things that makes the thing work.

Introduction

It is perhaps unusual for a computer technology to grow out of a social movement, but that is what this story is about. A small group of medical and computer people worked together for 15 years developing a computer technology to support a medical philosophy. The grant proposal of 1967 that provided our original funding set forth the goals that were to guide us:

We propose that (1) the medical record, utilizing a "problem oriented" approach, be the instrument whereby the following objectives can be implemented, and that (2) the technique of record keeping described in previous publications (Weed, 1964, 1968) be the basis for the beginning realization of these goals, and that (3) a real-time computer system be used to overcome the data distribution and time barriers that are insurmountable on a manual basis using conventional hospital and clinical medical records.

Objectives: To develop a system that will:

1) Facilitate good patient care by making immediately available (in minutes) to the individual physician a complete, updated list of problems on any patient and by providing simultaneously, as a unit, all the data in sequence (narrative, laboratory, etc.) pertinent to any of these problems.
2) Make possible epidemiological studies and other research endeavors in terms of problems, having all the data on any given problem immediately available.
3) Make possible a medical audit whereby the standards of care being provided for a given entity {e.g. hypertension) can be rapidly assessed because of the specific orientation of all the data.
4) Make possible a business audit to assess the physical, financial and time resources that go into the solution and management of a given problem. The need for a more organized, efficient and economical approach to the management of common medical and surgical disorders may then be documented.

These remained the objectives of the project throughout its history.

I met Lawrence L. Weed, M.D., in November of 1966. Weed had published his initial paper on a new organization of the medical record, "Medical Records, Patient Care and Medical Education" (Weed 1964). He was interested in exploring the implications of automating the new organization of the medical record. We decided to work together for six months and see if during that time a longer-term plan of collaboration could be worked out.

When we began our work, other medical record research groups took the dictated or written words of physicians and other medical personnel and manually entered them into a computer. We decided from the very beginning to interface the information originator directly to the computer and develop techniques and tools to facilitate this direct interface. We decided on this direction for a number of reasons: first, the time lag between the dictation and the entry of the medical information could have a negative impact on the patient care delivered; second, the use of the computer as a recall and structuring mechanism for medical knowledge and medical records could allow the medical personnel to operate from a universe that potentially could be much larger than what they could remember; and third, the data gathered in a specific section of the record could be consistent among all health practitioners entering it. The decision to directly interface the information originator brought us directly to the major areas of our research: the form of the human interface and the representation of medical knowledge within the computer.

Using computer technology to store and retrieve a medical record required that the medical knowledge entered into the computer have a rigorous structure. The medical community considered the notion of structuring a present illness, a medical history, or a physical examination tampering with the art of medicine. Developing outlines of the structure of the various elements of the medical record was only the beginning of the effort. We had to find internists, nurses, pharmacists, laboratory technicians, radiologists, and other medical personnel who were capable of externalizing the medical logic they had acquired and were willing to expend enormous mental effort to fill in the outlines with medical knowledge to make the system usable. The medical group in the Problem Oriented Medical Information Systems (PROMIS) Laboratory was such a group of people.

The structure of the problem-oriented medical record (POMR) served as our guide in defining the organization of all the data in the patient's medical record. The structure corresponds to four medical actions. The first is collection of a database; from this the problem list is formulated; for each active problem initial plans are written; and each problem is followed in progress notes. Without this structuring mechanism, our job would have been much more difficult because the traditional medical record has no logical structure but organizes data by its "source." For example, in the traditional record, all nursing notes are kept in one section of the record and laboratory reports are kept in another section.

The logical POMR structure requires that the same data be retrievable in many different ways. It is necessary, within a patient's record, to look at all progress notes for a single problem or all progress notes for the last 24 hours or a tabular array of specific physiologic parameters (a flowsheet). The requirements of the POMR were beyond typical database management systems of 20 years ago (and are beyond most of them today).

The medical record contains both narrative and numeric data. The narrative data of the medical record is variable in length and the numeric data has many forms. It was impossible to define a fixed length container or set of containers that could encompass all the data in the medical record. It was necessary to manage variable length text strings that could be retrieved in different orderings to different output devices. The numeric data contained in the medical record had hundreds of forms and had to be retrieved as part of a narrative note as well as part of a flowsheet. Both narrative and numeric data needed encoded information associated with it to facilitate the various retrievals and to make the data a legal medical record.

The initial decisions to interface the information originator directly to the computer system and to use the POMR to structure the record and provide an outline for the guidance within the computer system carried the medical and computer groups far into computer technology research and into advanced exploration of knowledge base development for the representation of medical knowledge.

A Search for the Correct Form for the Human-Computer Interface

INITIAL TOOLS FOR THE HUMAN-COMPUTER INTERFACE

The initial plan for the entry of medical record information was to use optical scan sheets and the IBM 1092-1093 programmed keyboards with an attached typewriter keyboard. Optical scan sheets were designed for the past medical history and systems review as well as for the physical examination. The programmed keyboards plus the attached typewriter keyboard were to be used for entry of all other information. The programmed keyboards accepted opaque plastic overlays or key mats placed over an array of 26 columns of 10 keys each, for a total of 260 labeled key locations. Each key mat could be removed and the programmed keyboard could sense the particular key mat being used, thereby identifying the type of entry into the system. Key mats were designed for each major medical subspecialty and were organized, minimizing the changing of key mats and maximizing the amount of information that could be entered from the key mat without having to use the typewriter keyboard. All information entered using the programmed keyboard was printed on the attached printer so that it could be verified before being stored in the computer.

During the first six months of 1967, several thousand past medical histories and systems reviews were processed, using two trained interviewers to fill out the optical scan sheets for entry into an IBM 1440 computer. The house staff at the Cleveland Metropolitan General Hospital was trained to fill out the physical examination scan sheets, and several hundred physical examinations were done. Great care had to be exercised in filling out the sheets or the optical scanner would reject them. The house staff had difficulty filling them out, and because of the large number of rejections, the use of the scan sheets was discontinued.

During the summer of 1967, Charles Burger, M.D., the designer of the key mats, tested the feasibility of using the programmed keyboards to enter a complete medical record. Burger's work was conclusive; the programmed keyboard presented the user with too much information, and the subsequent search time to find the proper key was excessive. The search time combined with mat changing time resulted in slow operation. A sample record took six hours to enter, a least double or triple the time required to write the original record. Scan sheets and programmed keyboards didn't solve the problem of effective entry of narrative data.

EXPANDED SEARCH FOR APPROPRIATE TECHNOLOGY

Because readily available off-the-shelf hardware was inadequate, we decided to investigate hardware and software approaches that could solve the difficult problem of an effective, facile human-computer interface whether or not the technology was commercially available or economically justifiable at that time.

We had a number of options open to us: We could develop the necessary hardware and software ourselves or we could find a group with experience in the relevant areas to leverage our own development efforts. I felt strongly that we should take advantage of whatever work other groups had done, so we began a search. IBM was promoting the use of the programmed keyboard that we already knew was not appropriate for us. Lee Stein was the first full-time computer person to join me at the Cleveland Metropolitan General Hospital. She had just come from the Hospital Computer Project, a joint Bolt, Beranek and Newman, Inc. (BBN) and Massachusetts General Hospital research project to study, develop, and evaluate computer techniques to serve hospitals in their information handling.

Stein and I visited BBN and spent a day with Paul Castleman. I decided that a time-shared Teletype terminal operating at 10 characters per second could not support the type of interface we needed. During this same period, Weed and I traveled to System Development Corporation and spent a number of days studying their efforts in the areas of database management, time-sharing and graphics, and light pen terminals. Again the technology did not fit my perception of what we needed. The massive graphics terminals seemed inappropriate for the hospital setting and the Teletype terminals were too noisy and slow.

During our expanded search for an appropriate technology, we met Robert Masters and Harlan Fretheim of the Research Division of Control Data Corporation. They jointly developed a cathode ray tube (CRT) terminal with a touch-sensitive screen, the Digiscribe, with software to support it. The software consisted of three interrelated programs: the Human Interface Program (HIP) managed user touch screen selections, the Selection Element Translator (SETRAN) managed the entry of new frames of information into the frame dictionary, and the SHORT operating system managed the execution of programs called from selections.

INITIAL USE OF A TOUCH SCREEN TERMINAL

From August of 1967 through February of 1968, we used a Digiscribe terminal connected over phone lines to a Control Data Corporation 160-A computer in Lansing, Michigan, at the St. Lawrence Hospital (Fig. 1). Initial versions of the Human Interface Program and the Selection Element Translator were available, and the new software tools required that the programmed keyboard key mats be restructured. This was to be the first of many conversions of the medical content to fit it into an expanded and more capable technology designed to enhance its use. In April of 1968, we installed a Control Data Corporation 1700 computer at Cleveland Metropolitan General Hospital.

FIGURE 1: Jan Schultz at a Digiscribe showing a visitor the system.

The Human Interface Program and the Digiscribe touch screen terminal allowed medical personnel to be directly interfaced to the computer system. An array of choices was displayed on the screen and users selected one by touching their finger to the screen over the selection. For each of the twenty selection areas, the Digiscribe generated a different character that the Human Interface Program could interpret. Based upon information in the frame, a branch to another frame could take place and new information could be displayed to the user (see Fig. 2). During normal selection processing, the keyboard was used for special functions such as erasing the last selection, aborting the current message, and quickly exiting the system. Selection processing used an internal stack of frames waiting to be displayed as well as user written programs that extended the normal selection processing. The result of a user's selections was a message displayed on the top three lines of the screen as well as other information not seen by the user but used by programs to interpret the selections. This unseen information included the frame number, the choice number within the frame, and internal parameters included with the selection on the frame. The internal parameters were used to code selections so that programs interpreted compact codes rather than alphanumeric data. The internal parameters were the coupling mechanism between the users making selections on the screen and the programs to store, manipulate, and retrieve the medical data.

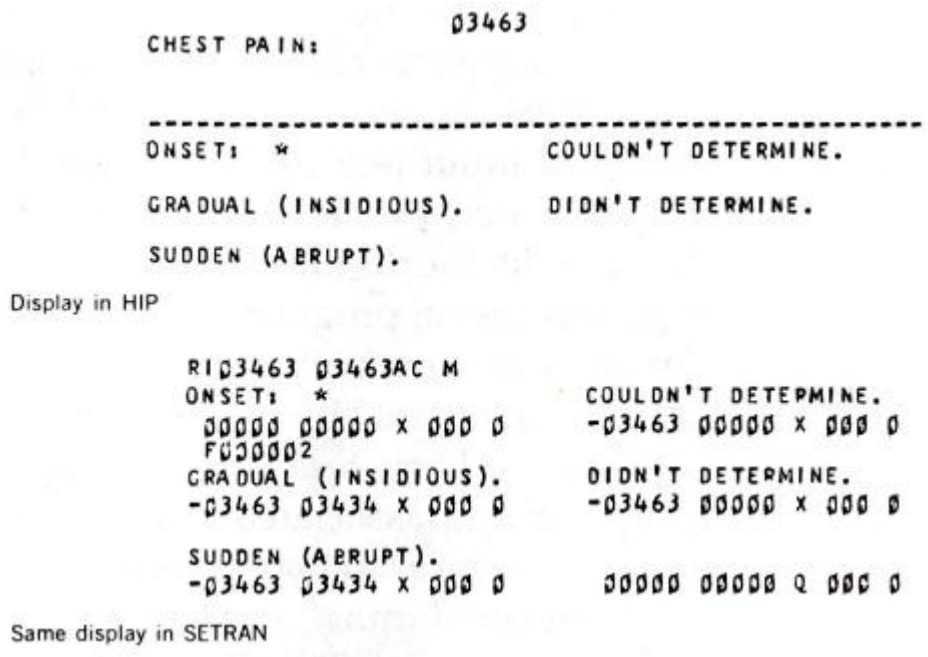

FIGURE 2: Message Generation Frame.

The Selection Element Translator was a program to edit frames. It was used to enter new frames and alter existing ones. The branching among the frames was also defined by the Selection Element Translator. The frames were organized into three types and contained selections that could fit on a

single screen. A frame was a physical unit defined by the screen size. The frame types were defined by the function they were to perform, and it was not possible to mix these functions among other frame types. The three functions were: the starting of new messages (and the termination of old ones), the generation of a message, and the display of textual information. The first type displayed an array of 7 x 2 choices, the second an array of 6 x 2 choices with the top three lines for the display of the generated message, and the third consisted of between 1 and 19 text lines.

We trained physicians and other personnel to use the Selection Element Translator so that the medical knowledge could be entered into the computer without a computer person acting as an intermediary. This ability was fundamental during the development of the system because of the massive number of displays that had to be entered.

The SHORT operating system supported multiple terminals as well as application programs operating in a multi-programming environment. The frames, application programs, and all medical data were mass-memory resident. Most selections made by a user required four disk accesses, and the station state data were resident in central memory only while the selections for that station were being processed. SHORT supported four classes of application programs: the first two allowed extensions to the Human Interface Program, one class allowed keyboard input (e.g., the Selection Element Translator), whereas the other did not; the next application program class ran in parallel with the user at the terminal and could communicate with the terminal (e.g., a retrieval program to display information to the CRT); the final class executed in the background after the user signed off the terminal. The data generated by the user could be passed to each program class.

The hardware base consisted of a Control Data Corporation 1700 computer and associated peripherals. The CDC 1700 had 32k 16 bit words of central memory with 1.1 microsecond cycle time, and the CPU executed most instructions in 2.2 microseconds. The mass-memory was CDC 854 disk drives, each holding 7.2 million bytes of storage and each with an average seek time of 70 milliseconds. The Digiscribes were connected to the CPU through a display controller with a delay line for each terminal. The maximum distance between the display controller and a terminal was 1000 feet. The transfer rate between the CPU and the display controller was 50,000 characters per second. The CRT screen had 20 lines of 50 characters each. The Digiscribe consisted of transparent, conductive pads on the CRT screen and the associated electronics sensed which of the 20 pads were touched. Each pad covered the left or right 25 characters of every other line.

INITIAL MEDICAL CONTENT DEVELOPMENT

Developing a system that enabled a physician, nurse, or other medical personnel to enter narrative medical information using a touch screen as the primary interface was a major challenge. Stephen V. Cantrill (Fig. 3), an IBM Fellow during the summer of 1967 (before starting Case Western Reserve Medical School), analyzed a large number of manually recorded cardiovascular problems and defined 90 frames that together could be used to enter most cardiovascular problems. Also, during the same time period, the frames defining present illnesses for GI and liver problems were written.

FIGURE 3: Steve Cantrill showing off the Terminet printer. Photo by Jim Wanner.

Because of the limitations of using the optical scan sheets for patient questionnaires (which included no branching capabilities, interviewer administration, and a high rejection rate), we investigated the feasibility of having the patients use the touch screen for eliciting historical information. Werner Slack, M.D., a medical researcher at the University of Wisconsin using a LINC computer, had demonstrated that directly interfacing the patient with a CRT terminal for data collection was feasible for a selected patient population. We questioned whether we could do it routinely for all patients. We developed two short questionnaires for self-administration and found that except for people with poor eyesight, with poor reading skills, or with limited ability to read English, the questionnaires could be used easily. Children down to the age of 10 years could also use the touch screen terminals to answer the history questions.

INITIAL ELECTRONIC RECORD DEVELOPMENT

The programs to store and retrieve all types of medical data were evolving in parallel with the experimental entry of the narrative medical data using frames. All medical data that we thought could be effectively stored in an electronic record were classified as either narrative or numeric. Narrative data was further classified as resulting directly from selections or resulting from self-administered patient questionnaires (translated data). There were three types of data stored in the medical record: direct narrative, translated narrative, and numeric data. Keith G. Morgan, a Ph.D. student in the Engineering Program at Case Western Reserve University, joined us at Cleveland Metropolitan General Hospital. Morgan developed a program to store numeric data and wrote retrieval programs to display the numeric data in flowsheet form. Cantrill developed a program to translate a questionnaire's responses into narrative text for storage into the patient's record. He also developed a program similar to the Selection Element Translator for the entry and modification of dictionary

elements used in the translation process. Lee Stein developed a program to transform the user's selections into narrative text for storage into the patient's record. I developed routines to assist in the display of the stored data. One routine, called FORMAT, interpreted codes stored with the narrative data so that the internal form of the record data was output device independent.

The ability of the Human Interface Program to couple narrative selections with internal parameters stored in the frame allowed the frame developer to enter the parameters needed to encode the data and to define the data's output format at the same time the content in the frame was entered.

INITIAL PHASE OF DEVELOPMENT

The initial phase of the development focused on the correct form for the human interface. Other options were tried, but very early on we settled on a CRT with a touch screen as the ideal interface for both medical personnel and the patient. At the time we did not fully realize that the interface was effective not only because of the touch screen, but because of the high data fate of 50,000 characters per second between the CPU and the terminal and the system's ability to process most selections in under 0.5 of a second.

In Search of the Right Development Environment, We Move to Vermont

In July of 1969, a group of five families moved from Cleveland, Ohio, to Burlington, Vermont, and PROMIS Laboratory was formed. One of the original goals for the automation of the POMR was the integration of the patient's bill with the clinical medical record. The hospital in Cleveland was a county hospital, and while we were moving from the initial phase of our research to a phase requiring access to more of the hospital, the hospital's data processing department was shifted to the county data processing department. This meant that there would be no opportunity to integrate the electronic medical record with the patient's bill and the hospital's administrative operation. We began looking for an appropriate site to continue our research.

The University of Vermont's Medical School and Medical Center Hospital offered us the opportunity we were looking for: a teaching hospital closely associated with a University through which our funding could continue and an assurance that the data processing department at the hospital was progressive and willing to work with us.

The Medical Knowledge Database in the Early 1970s

The medical database consisted of two large bodies of knowledge: knowledge that traditionally resided in medical libraries, journals, and generic medical knowledge was housed in frames; patient-specific knowledge resided in electronic patient records. This section will describe the frame-based generic medical knowledge.

Getting the initial PROMIS system operational on a medical ward in 1970 required a massive amount of medical content to be available on the frames for selection by the medical users. Drs. Larry Weed and his wife Laura B. Weed worked together translating medical content into a form that could be displayed for a medical user. The first step was to create a skeleton structure, as a first approximation, that was a synthesis of remembered experience, a review of classical textbook descriptions of disease, and a review of the appropriate current literature. This process was completed in a year. Once the initial framework was complete, consultants in various subspecialties were used to help continue the building process.

DRUG INFORMATION FRAMES

The *drug information frames* were constructed by George Nelson, M.D., Genevieve Gilroy, registered pharmacist (Fig. 4), and Brian Ellinoy, Pharm. D. The drug information frames provided the medical user using the system with the basic facts about commonly used drugs. The guide for entry of the initial information was focused on the common drugs, since most adverse reactions occurred with them. One hundred and thirty of the most commonly used drugs at the Medical Center Hospital of Vermont were compiled, and by 1972 over 180 separate drugs were represented in the system. The information was largely abstracted from standard textbooks and articles in pharmacology, with supplementation from the current literature. The drug data was organized in displays entitled "Check Problem List for," "Side Effects to Watch for," "Drug and Test Interactions," "Usual Dosage," "Mechanism of Action," and "Metabolism and Excretion." All information was reviewed by other physicians interested in clinical therapeutics.

FIGURE 4: Genevieve Gilroy and George Nelson discussing drug information at a CDC 211 terminal attached to the CDC 1700 system. Note the touch strips on the CRT screen and the antenna in front of the keyboard. Photo by Jim Wanner.

DIAGNOSTIC PROCESS

The initial system was used as an aid not only in making therapeutic decisions, but also in arriving at diagnostic plans. George Nelson did the initial work on the diagnostic plans. This was not diagnosis by computer, but it used the computer's recall ability and the electronic medical records to assist the physician in the diagnostic process. The first step in the process was the creation of problem formulation sequences for most medical problems and the association of each problem with a unique number, called the *problem plan number*. Plans were broken up into *get more information* and *treatment*. The *get more information* section was further broken up into *for more data base*, *diagnostic process*, and *for management*. The *diagnostic process* was further subdivided into three sections. A single problem plan number could have information about six or seven aspects of the plan.

By August of 1973, there were problem-specific diagnostic and therapeutic plans for over 600 problems covering a wide range of medical problems. The plans were audited by outside medical authorities to assure the safety and appropriateness of the decision pathways. Most of the frames were documented by reference citations and the citations were available in the Medical School Library. We did not have enough mass-memory space available to keep the citations on-line. By this time the drug displays had been expanded to include over 300 individual agents.

RADIOLOGY REPORTING

The initial system also included frame-based medical knowledge for use in support of the radiology department. Peter Dietrich, M.D., a radiologist, developed the medical content in this area. The x-ray plans for many problems formulated in the computer were built. A structured reporting format

specific to both problem and radiologic procedure was developed. Reporting sequences for over 150 problem-procedure pairs were built by October of 1972.

PROBLEM PLAN NUMBERS AND TABLES

In order to accommodate the problem-specific information in the system, a new container for medical knowledge was necessary. The information in the container was never seen directly by the user, but was needed to collect together all the frames that related to one problem plan number. The new container was called the *branching information table*, and it had slots for each type of problem-specific information. Each slot contained a frame number that could be put into the branch stack that the Human Interface Program used, so that if a specific selection were made on a frame and the patient's problem were known, then the problem plan number could be used to look up the correct branching information table slot for the selected problem. The branching information table was the first of many tables that would come to exist as part of the medical knowledge database.

The *table* was the container for all machine-readable data related to a medically defined problem or procedure or other entity such as a system user or terminal. Each table had a unique address or code. Most of the data in a table was not visible to the medical user, but was used by programs to perform actions or do special display branching.

An Operational PROMIS on Medical and Gynecological Wards

The PROMIS system replaced paper records with a problem-oriented electronic computer record for six months on a general medical ward and nearly four years on a gynecology ward in the Medical Center Hospital of Vermont. More than 3000 patients were followed using this system during its operation, from July of 1970 through November of 1974. More than 500 individuals (internists, nurses, medical students, house staff, radiologists, pharmacists, social workers, ward secretaries, dieticians, and laboratory technicians) used PROMIS as the sole mechanism to add to and retrieve information in patient records. The prototype PROMIS system demonstrated solutions to many general problems seen in other computer applications. Through redundant hardware, careful maintenance, and good diagnostic tools, the system was made available to users more than 99.6 percent of the time, within a scheduled 24-hour, seven-day week operation. The system was designed for fast response to facilitate use, and would respond to a user's selection by presenting another screen within 600 milliseconds for over 50 percent of the selections. Response was both rapid and comprehensive as the terminal made available the patients' complete records as well as more than 30,000 displays of medical information for guidance.

Eligible personnel throughout the institution were allowed access to the electronic record. Terminals on the ward, in the operating suite, in the pharmacy, in the clinical laboratory, and in the department of radiology demonstrated the synergism possible using an electronic record.

The overall goal during this time period was getting PROMIS operational on a single hospital ward and analyzing the operational hardware and software system in enough detail to allow realistic estimates for the size, cost, and system structure necessary to implement PROMIS throughout a hospital.

THE SOFTWARE TO SUPPORT AN OPERATIONAL PROMIS

The software developed to support the operational PROMIS (Fig. 5) included the SHORT operating system, the Human Interface Program, and the Selection Element Translator, supplied by CDC. The Store program transformed the data generated by the Human Interface Program into patient record and other system files. The Retrieve program abstracted information from the patient's record or other system files and displayed the abstracted data in the form of a narrative report or a flowsheet. When all of the application programs were written for the CDC 1700 system, there were over 250,000 lines of assembly language code (Schultz, Cantrill, Morgan 1971).

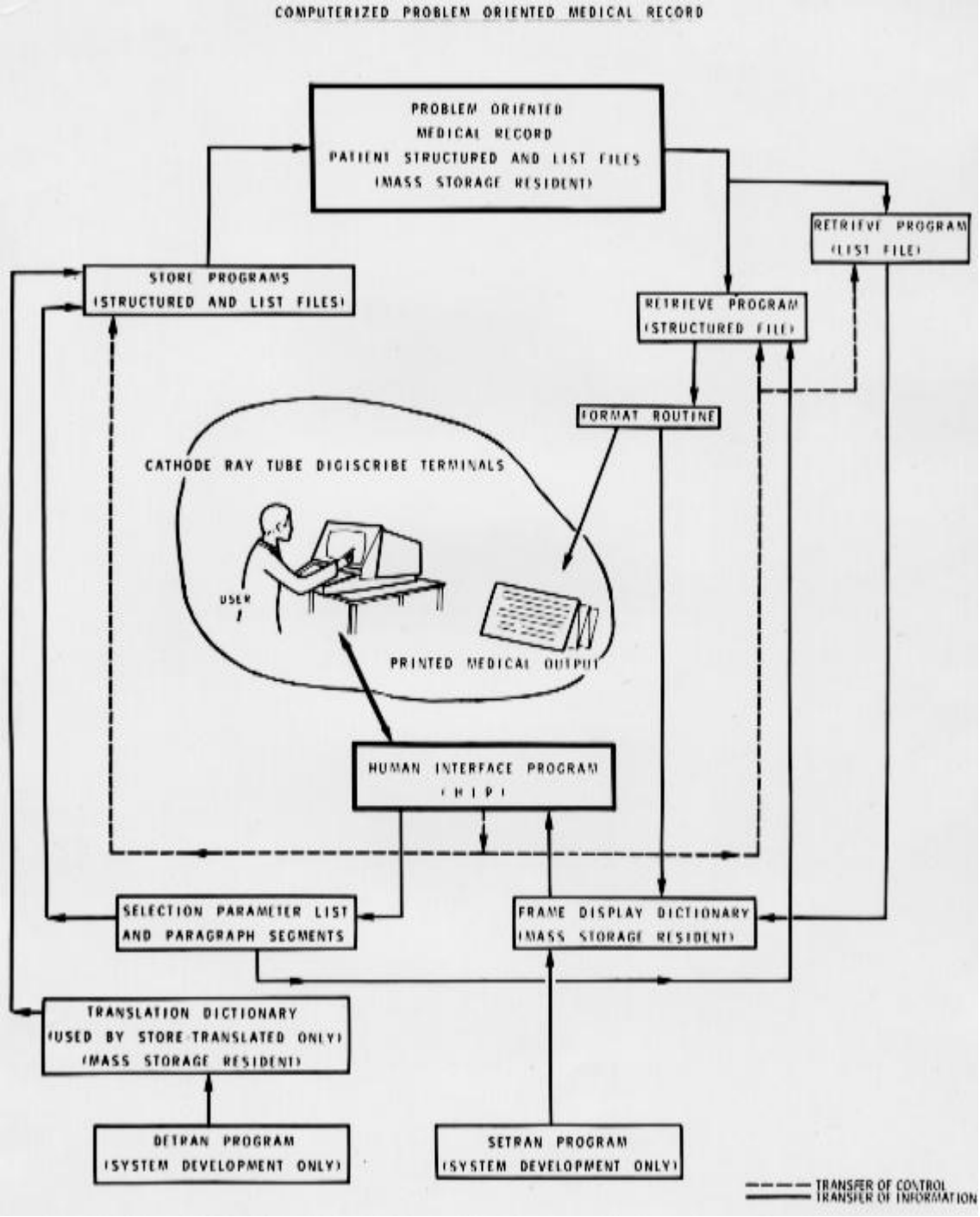

FIGURE 5: The elements of the application software.

THE FORM OF THE DATA WITHIN THE ELECTRONIC RECORD

An individual patient's electronic record was mass-memory resident and required a structure that facilitated its manipulation while minimizing the number of mass-memory accesses required to

perform the manipulation. An individual record consisted of one fixed length index block and a variable number of fixed length data blocks. The index block was an index to all the data in the data blocks and was accessible using the patient's system identification number. The index block contained a variable length list of pointers to the data blocks along with other information to identify the type of data in the data block. To access data in the patient's record, a search was performed through the index's data block pointers looking for those pointers that satisfied the retrieval request. For any single data item retrieval, two mass-memory accesses were required.

The internal form for a data item within the electronic record depended upon whether the item was narrative or numeric. For narrative data, the internal parameters associated with the selections were interpreted as format codes to be used to define the retrieval format. For numeric data, the internal parameters associated with the selections were interpreted as data types and each data type was stored in a separate fixed length field. Consider the following example: a temperature's internal parameters were an internal code number, a time (10:23), a date (February 23, 1970), a title (Temperature), a numeric value (38) and a numeric qualifier (degrees Centigrade).

THE SELECTABLE LIST FILES

To facilitate access to rapidly changing patient record information, it was necessary to present the patient information on displays for selection. A list of patients on a ward or the list of current medications for a patient are examples. Displaying the lists for selection required that the data to be displayed be readily available so that it could be retrieved quickly. The selectable list files were an abstraction of data stored in the patient's record to facilitate the rapid retrieval of selectable lists.

THE MEDICAL GUIDANCE TO SUPPORT AN OPERATIONAL PROMIS

When health care providers sat down to use the terminal, they first chose the name of a patient presented on their terminal screen and then selected one of the sections of the medical record—the database, problem formulation, plans, and progress notes—in order to add to or retrieve from that part of the patient's record.

Database

The nurse entered the patient's chief complaint, patient profile, and the general appearance and vital sign components of the physical examination. The patients, after only two to four minutes of instruction, entered their own review-of-systems by answering a series of up to 275 yes/no and multiple-choice questions. Only the positive responses were stored in the record. The review-of-systems was then retrieved and reviewed by medical students and house officers. Positive responses were developed into present illnesses by touching selections specific for each symptom. The physical examination would then be entered using up to 1410 frames, the number actually used being a variable related to the patient's physical status.

Problem Formulation

After completion of the database, the physician formulated the patient's problems by making a series of display selections. Each problem would be stated at its level of definition, ranging from a symptom to a firm diagnosis. The displays facilitated further description of problems by guiding the user—frequently providing them with criteria to help use consistently precise language. These criteria or definitions were taken from the literature with a library reference citation noted on the display. The system facilitated easy updating of any problem when new information gave added insight. Problems could be added or their status changed (e.g., from active to inactive) at any time.

Plans

For many problems in the system, each component of the plan was individualized. The system

contained displays based on the current literature detailing how to evaluate, treat, and follow the problem. These displays removed reliance on memory and aimed toward providing a new intern the ability to evaluate the problem with a degree of accuracy and completeness comparable to a sub-specialist. The displays did not constrain the users; they were free to order whatever they pleased, but the displays served to guide them. More than 600 specific disease entities had some degree of problem-specific information in the system.

Progress Notes

Progress notes are all notes written about a problem after the initial plans and include those written by physicians, nurses, social workers, chaplains, and others. All progress notes were entered by first selecting the appropriate problem on the terminal screen from the patient's list of problems. The symptomatic and objective sections took the user to frames specific for the body system of the problem. The nurses were encouraged to enter progress notes according to the severity and rate of change of each problem of each patient under their care. They used the same displays as the physicians, although some displays were used more by nurses than by physicians. Operative notes, radiology reports, clinical laboratory reports, social service consultations, and specialist's consultations were also entered as progress notes under the problem they addressed.

In September of 1974 there were approximately 30,000 frames in the system. A breakdown showed that 6,000 of these were database frames, 4,000 were problem list frames, 7,000 were plans frames, 8,000 were progress notes frames, superstructure and others made up the remaining 5,000 frames.

DATA RETRIEVAL

The above four sections comprise the basic patient record. This data was kept in electronic form in the computer system during a patient's hospitalization. There were no written records maintained in parallel. The updates to the record were printed each day and were available on the ward. Patient information could be retrieved at any time at any terminal by an appropriately identified medical user. Since the record was in electronic form, data could be retrieved in many ways. Information could be retrieved on a specific problem, on all problems chronologically, or in flowsheet format. Current outstanding drug orders, investigations, and accumulated charges were also available by problem. The problem of absent or lost records was nonexistent. The records were all legible. Any number of users could simultaneously retrieve from or add to a record.

PROBLEM-ORIENTED LIST OF CHARGES

One of our initial goals, the creation of a problem-oriented list of charges, was accomplished. We compared a hospital bill with a list of charges calculated from the chargeable items in the electronic record. We also presented a patient's problem list with the charges associated with each problem so that medical users and administrators could see the cost of treating each problem. We discovered that the problem-oriented list of charges created from the electronic record was different from the hospital's bill since it represented everything that was recorded in the record that could be charged, and not just those things that had a charge slip written for them.

THE HARDWARE TO SUPPORT AN OPERATIONAL PROMIS

The hardware base consisted of two Control Data Corporation 1700 computers and associated peripherals (Fig. 6). Each central processing unit had 32k words of central memory and had seven disk drives with 7.2 million bytes capacity each. There were 22 local terminals and 8 remote terminals. Remote terminals operated at 4,800 bits per second and contained their own delay line refresh memory. The remote terminal multiplexor communicated full duplex at 4,800 bits per second. A

modem at each end of a C2 conditioned, leased 4-wire phone line was used to connect the remote terminals to the terminal multiplexor. The 4,800 bps modems cost over $10,000 each and had to be manually equalized to the telephone line.

FIGURE 6 Ernie Preiss working on the CDC 1700.

INFORMATION UTILITY

The continuous operation of the system required the development of a reliable, backed-up hardware base that the users considered to be an information utility. One of the major elements that contributed to keeping the system in continuous operation was our ability to see patterns in the occurrence of various system problems. Whenever a problem occurred that affected the operation of the system or more information on the problem was available, we described the problem in problem-oriented fashion and posted it so that periodically we could discuss the current operational problems. We kept paper logs of all operational problems since there was not enough mass-memory space available to keep the data on-line (Fig. 7). Eventually, when more mass-memory was available, we kept the problem statements on-line. There were four major problem categories, divided into hardware, software, architectural relationships, and undefined. Each category was divided into maintenance, operation, environment, financing, personnel, development, and education. Problems in the hardware, central processor, and peripherals operation category ranged from "core parity errors" to "idle time low for unknown reasons." Problems in the hardware environment category included "Ernie's tools disappearing" and "CPU room floor dirty and place a mess." The total log for this time period included over 500 active problems. One of our mottos for the period was to "let the problem lead us," and we had enough problems; the only question was where they were to lead us and which problem would be the leader.

FIGURE 7 This is 50 megabytes of mass storage in 1970.

The National Library of Displays

The medical group within PROMIS Laboratory was formed to create and maintain the thousands of computer displays. The medical content was viewed as a textbook or encyclopedia organized and available according to the patient's problems. As Dr. Weed wrote (Weed 1972):

The opportunity now exists in modern medicine for making available to health care personnel tools which will permit them to perform with excellence without depending on an encyclopedic memory... It is therefore an appropriate goal to seek to make available to all health care personnel a tool which simplifies to a uniformly excellent level their initiative, memory, and ability to create and to execute reasoned and disciplined plans... The goal can be accomplished in large part if the tool has built into it the parameters of guidance and currency of information required by health care personnel as they perform and record their work. Then, individual human memory and initiative will cease to be the critical link between all we know and all that must be done.

THE PROCESS OF MEDICAL CONTENT DEVELOPMENT

The process of creating the medical content involved much more than just entering the medical knowledge onto the frames. Prior to the release of the medical content for on-line use, a rigorous in-house audit was completed. Outside experts in the relevant medical subspecialties were invited to Burlington for a three to five day period, to audit each frame sequence for safety, accuracy, currency, and completeness. The audit comments were then reviewed by the author of the frame sequence, and appropriate changes were made. Each step of the process was documented. Finally a check of the branching was done to verify that the sequence worked as planned.

PRELIMINARY STUDY BY NATIONAL BUREAU OF STANDARDS

The process outlined above was limited by the resources of the medical group of PROMIS Laboratory. It became clear that prior to the widespread distribution of systems with the characteristics of PROMIS, a national body should be created to assume the responsibility for the review and distribution process. This would ensure that the systems represented accepted standards in medical care as well as computer performance. The National Bureau of Standards in conjunction with the Health Services and Mental Health Administration sponsored a study directed at identifying actions appropriate for facilitating the widespread use and maintenance of PROMIS. Melvin Conway, Ph.D., was hired by the National Bureau of Standards to do a preliminary study.

The report from the study was available in October of 1972 and was an initial planning document. It raised many issues: display library accreditation, standardization, distribution, quality control, the administrative tools required by the library, certification in the field of the record-maintenance systems that are using the displays, and controlled evolution of the entire process. The report defined terminology and concepts that we incorporated into our thinking and ultimately into our evolving system. It defined generic system specifications that set forth criteria for initial certification for record-maintenance systems so that system vendors could design to the specification. It defined under this a system line, which is a definition of a class of concrete record-maintenance systems, and under this a system instance, which is an individual medical record-maintenance system.

A parallel, three-level hierarchy of definitions existed for the collection of displays. The generic library is the primary standard resident in the national repository from which all specific display collections residing in system instances are drawn. A sub-library is a subset of the generic library to allow distribution of non-identical display libraries. A library instance is a machine-readable data set of a

sub-library intended to reside in one system instance. And finally, a library instance update is a portion of a library instance that is sent out to a system instance when only part of its display collection needs to be changed.

RELEASE OF FRAME AND TABLE LIBRARY

PROMIS Laboratory was committed to the formation of a National Library of Displays. John M. Nelson, M.D., coordinated this activity. We resisted efforts by the government and private industry to get early releases of the display library; we feared if multiple copies were released, it would become impossible to provide updates since the mechanisms for creating sub-libraries and library instances had yet to be worked out. The merging of already released library instances with a library instance update was a difficult problem. The tools to perform the library instance update were not in place until 1979.

Our National Center for Health Services Research contract in 1976 required that a data description language be developed to externalize the frame and table library in a form that was machine independent. We made available on magnetic tape, yearly, the complete library as well as a copy of all of the computer software from 1976 until the last contract expired in September of 1981.

Expandable and Exportable Architecture Is Defined

Besides the continued operation of PROMIS in the hospital and the entry of new medical knowledge, a major effort was undertaken starting in the spring of 1973 to define an architecture that could serve a full hospital as well as be exported to other medical settings and different size institutions. Wesley Clark and Mel Conway were hired by the National Center for Health Services Research to assist us in this task. The operational CDC 1700 served as a resource for defining the subsystems for a new architecture. The CDC 1700 system consisted of three subsystems: The human interface manager had both hardware and software elements. The station hardware included the cathode ray tube terminal, the Digiscribe unit to make the screen touch sensitive, the hardware logic to make the terminal behave in the expected way, and the display controller or multiplexor that interfaced the terminal and the CPU. The software to control the human interface interpreted the frames, concatenated the selections together to generate English narrative, determined the next frame and presented it to the user for selection, and then reinitiated the cycle. The patient record manager included the mass-memory to store the patient's record as well as the software to update, retrieve, and take patient data on and off-line. The frame manager included the mass-memory to store the frames and tables, and the software to create and modify them.

SUBSYSTEMS CHARACTERISTICS

Each of the subsystems had characteristics that were used to help us determine the means of instrumentation. These characteristics were determined by the way the system was implemented and by the way it was used within the hospital. The major characteristics of each sub-system are described below.

Human Interface Manager

This subsystem had to be extremely responsive. We defined extremely responsive to mean that the time from user selection to complete display of the next screen's data was less than 0.25 of a second, 70 percent of the time. This responsiveness was necessary to facilitate the effective interface of professional non-computer trained people to a functionally complex computer system. The human engineering of the touch screen is essential for the interface to work well. The data rate between the frame manager, the human interface manager, and the station must be high. Only 50 milliseconds was allocated for the transfer time between the frame manager and the station. The displays used to generate data were sparse, with an average of 6 choices used out of a maximum of 14. The combination of the highly responsive system with sparse displays facilitates pattern recognition and rapid interaction by the user. Unlike other computer systems where the function of the human interface was to get to a computation, in our case the interface was the computation.

Patient Record Manager

This subsystem must be extremely reliable so that patient data is never lost and so the many types of retrieval requests for individual patient records as well as for groups of records are handled rapidly. Individual records ranged in size from 10,000 to 100,000 characters (or more) with an average of 6000 characters of data added daily. The access time to one element of a patient's record was 65 milliseconds, and it took from 2 to 200 accesses for most retrievals. An average storage into a single patient record required 6 accesses to the record file and 20 to 100 accesses to the input data generated by the human interface program.

Frame Manager

This subsystem contained 30,000 frames and potentially could contain two to three times that number. Over 18 million characters of storage were required, and the average number of characters

per frame was 600. The 600 characters included not only data to be displayed to the user, but other branching and internal parameters. The internal form was not densely packaged; groups of fixed length fields were present whether filled or not. Two mass-memory accesses were required to retrieve one frame. Twenty percent of all accesses were to two percent of the frames, so that a faster access time storage media for a small number of frames could have an impact on the throughput of this sub-system.

POTENTIAL EXPANSION OF THE CDC 1700 SYSTEM

The CDC 1700 system could not be expanded to operate the total hospital, not only because it was technologically disadvantaged, but because the system software had many limitations that could only be changed with a major software restructuring. It is valuable to consider the types of walls present in that system. They became the guide to a redevelopment effort. Each wall was a design decision in the Human Interface Program or the patient record structure. The Human Interface Program's design restricted the maximum number of selections that could be made within one generated unit so that frame sequences had to be changed to keep the generated unit from overflowing; the frame library could contain a maximum of only 32,000 frames, and we were very close to overflowing it. The structure of the patient record restricted the size of the patient record index to a single mass-memory block so that some very long patient stays would overflow the block requiring the starting of a new record; the numeric information for one data item in the record had a fixed maximum size; a maximum of 144 patient records could be on-line at one time; the patient's problem and order lists had a maximum size so they had to be monitored and cleaned up as some of them reached maximum size; the numeric and narrative data were stored in separate parts of the record and retrieval functions that operated on narrative data couldn't be applied to numeric data and conversely. This made it impossible to retrieve a flowsheet of narrative data.

INCORPORATE EXPANDED FUNCTIONS

It was also necessary to incorporate expanded functions into the system's operation. We wanted to be able to fully exploit the electronic record by allowing its full access in all the supporting areas of the hospital; by performing data compression of the electronic record; by making available at all times the patient's problem list and current medications for all patients in our population; by developing a patient record that could span multiple admissions and serve for the patient's lifetime; and finally, by linking the medical data in the record with the patient's financial data.

SCALE TO SUPPORT TOTAL HOSPITAL

The new system had to be able to support the total hospital. The size of the system was determined by the number of active patients in the system at one time and the total served by the hospital, by the size of an individual patient's record, and by the number of terminals active at one time.

The Medical Center Hospital of Vermont in 1973 handled 20,000 admissions and 120,000 outpatient visits per year. There were between 25 and 85 admissions per day and 10,000 outpatient visits per month.

The size of a single patient admission was determined by looking at a "typical" six-day period of system operation and extrapolating. We used figures for the number of admissions per year and outpatient visits per year to arrive at the amount of patient record growth expected in one year. It was 1.24 billion characters or the equivalent of 4 double density IBM 3330s (the measuring stick of that era).

The number of terminals for the hospital was calculated based upon 4 terminals per 20 bed ward and

7 terminals per 34 bed ward and terminals in all the supporting areas. The total number was 200.

ARCHITECTURAL CHARACTERISTICS ARE DEFINED

Given the functional characteristics of the CDC 1700 system and the characteristics of the site for installation, an architecture for a PROMIS system was defined. The major elements of the PROMIS architecture were redundancy of hardware elements for reliable service, guaranteed responsiveness with minimal sensitivity to load, and access to any patient's record from any terminal within the system. The architecture had to support the locational diversity of the health care system since health care is not practiced in one geographical site, yet it was important to allow communication among the sites.

TWO REMAINING ISSUES

Besides the elements for the architecture listed above, there were two issues that had to be explored before a final decision on an architecture could be made: first, should the subsystems be functionally partitioned into separate hardware elements, and second, should the patient record files be centralized or distributed?

One potential architecture was a network of minicomputers. Each node of the network was to be one minicomputer that would handle between 10 and 30 terminals. The number of terminals per node would be determined by requiring a system response of less than 0.25 of a second 70 percent of the time. The nodes would each contain the three subsystems. A communications medium would allow the nodes in the network to communicate. The patient records that were normally accessed by the terminals connected to one node would be contained in that node. Other nodes' access to a patient record would be on-demand and would require a transfer across the network. For storage into a record, packets of data would be shipped to the node that contained the patient's record.

Another proposed architecture was a functional partitioning of each subsystem into separate hardware elements. There would be frame engines containing the frames in a read-only form. Each frame engine would be a small minicomputer with mass-memory to contain the frame library and communication ports so that the human interface subsystems could communicate with it. The human interface subsystems would contain no mass-memory and would multiplex many terminals. Each human interface subsystem would have communication ports for connections to both a frame engine as well as a record processor. The record processor would be fully duplexed minicomputers with communication ports for connection to the human interface subsystems and to remotely located printers.

The issue of whether the patient record file should be centralized or distributed was posed in the following way: Can we build a system that can serve New York City as easily as it serves the City of Burlington and the State of Vermont? Ultimately there would have to be a medical record network serving a very large region; the question was, Shall we learn the problems of building such a network while it is small or wait and solve the problems posed by such a network only when it is no longer feasible to keep the data files centralized?

A paper simulation was done of the system data traffic among the subsystems, assuming the architectural model of a network of six minicomputers with each minicomputer managing 50 terminals and with intra-node traffic for a distributed patient record file being 25 percent of all accesses. The average total data rate was 42,000 characters per second; assuming a burst rate of ten times that rate and assuming maximum intra-node traffic for the patient record files, the data flow would require only 25 percent of a million word per second I/O bus. The remaining bus capacity would be available for instruction execution, data pool swapping, and program loading. Given the

system data traffic requirements and the type of hardware available at the time, there was no need to segment the functions into separate hardware units.

FINAL DECISION

Our decision was to define an architecture consisting of a network of minicomputer nodes with each node containing the three subsystems. The patient record files would be distributed across the network. To reduce network traffic associated with access to these files, a medical record placement scheme was used to maximize the probability that the patient's record and the provider's terminal were local to one another on a network node.

Specify Technology to Support the New Architecture

With the architecture defined by the end of 1973, our next year's jobs were set out before us. A minicomputer, terminal, inter- and intra-node communications bus as well as a programming language had to be specified and selected for the new PROMIS system. We had been looking for a hardware person to round out the PROMIS computer group when James Wanner (Fig. 8), an astronomer and mechanical engineer with digital electronics experience, walked through the door. His first task was the writing of a terminal specification. We decided to use MOS technology where it could be applied. MOS was "mostly off the shelf," and those items we could not find on the shelf we tried to put on the shelf. Detailed specifications were written for all of the items and we set things in motion to begin the procurement process.

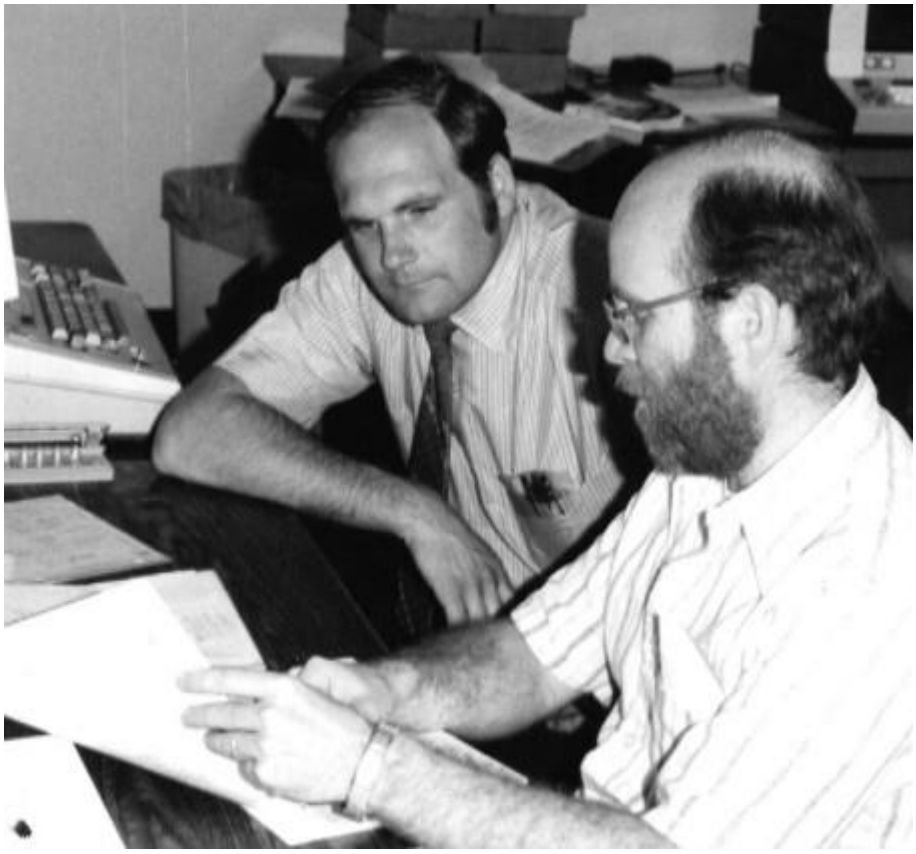

FIGURE 8 Jim Wanner and Ernie Preiss discuss the Megadata terminal specification. Note the prototype Megadata terminal on the table in the right rear.

MINICOMPUTER SPECIFICATIONS

The *minicomputer* had the following specifications: central memory of at least 128k, 16 bit or greater words with a cycle time of one microsecond or less, and direct memory access to peripherals; the environment protection had to include power failure restart circuitry and error checking on all data

paths in the system; multi-level interrupts; memory protection consistent with any memory mapping or multi programming executive; mean time between failure of less than one processor failure per year. Mass-memory was divided into 10-millisecond fast access for 2 million characters and 65-millisecond access for 200 million characters. Communications subsystem (four wire or phone) must support printers and inter-computer communications at 50,000 bits per second or faster. Up to 50 local touch screen terminals operating at 16,000 characters per second or faster had to be able to be connected to the system. Other standard peripherals had to be available.

TOUCH SCREEN SPECIFICATIONS

The *touch screen terminal* had the following specifications: 8 1/2 inches high by 11 inches wide screen; two display modes (the first having 24 lines by 80 characters each, with 12 lines aligned under the touch-pads on the screen, and the second having 36 lines by 100 characters each with the bottom line aligned with the bottom line of touch-pads on the screen); alphabetic characters shall appear in at least three distinct ways; if a cathode ray tube is used, the display shall be flicker-free for all users; the touch pad screen shall have an array of transparent touch pads each covering a fixed block of characters in a displayed page of text. The pads shall respond to the operator's finger touch, independent of body position. When a pad is touched, the text lying under the pad shall be emphasized. This emphasis will continue until a new display is received by the terminal. The bottom line of screen display text shall have 10 touch pads, each pad covering 7 characters, with a one-character-wide space between the pads. Eleven additional lines of text, on approximately 5/8-inch centers, shall have four touch-pads each covering 19 characters of text. Parallax errors between text and touch-pad positions shall be minimized by mounting the pads as close as possible to the text. Data flow into each terminal shall be at the rate of 15,000 characters per second, or faster. Data flow from each terminal shall proceed at a rate of at least 15,000 characters per second, or less if appropriate to the terminal controller subsystem. The terminal interface subsystem shall support at least 24 terminals in apparent concurrent active use. Each terminal shall have access to more than one central processor unit, for use as backup on CPU failure. Portable terminals shall be provided by means of terminals mounted on wheeled carts with single plug interface to the system and taps at various locations.

COMMUNICATIONS SUBSYSTEM SPECIFICATIONS

The *communications subsystem* had to be able to support up to 50 touch screen terminals operating at 15,000 characters per second full duplex, up to 20 printers operating at 9600 characters per second, and at least one inter-computer connection operating at 50,000 bits per second.

PROGRAMMING LANGUAGE SPECIFICATION

The *programming language* specification included the following description. A high level language is desirable, but not at the expense of execution speed or mass-memory access efficiency. The language includes character manipulation instructions. It is machine independent so that the application programs do not have to be rewritten when the machine is changed and so that the same programs can run on multiple machines. The language must be transparent to the operating system. The language must incorporate the ideas of structured programming. There can be separate languages for application programming and systems programming. The original assembly language programs were very efficient, very difficult to change, and had very little documentation as part of the program; we must do better this time around!

The Architectural Elements

MINICOMPUTER PROCUREMENT

Request for proposals was sent to eight computer manufacturers. Four offered no bid. IBM said they had a "hole" in their product line and that a network of their computers would be too expensive. Digital recommended an 11/45, which met the hardware requirements, but would not make information available on their software. Although they offered to sell us hardware, they made no bid on our request for a proposal. PRIME and INTERDATA made no bid.

We received four proposals. The ModComp, Inc., proposal met the hardware specifications, but the system they proposed, the ModComp IV, was not yet available for testing. Control Data Corporation sent an extensive proposal; William Norris, the CDC President, delivered it and spent the day at PROMIS Laboratory discussing the continuation of our long-term relationship. CDC proposed the most interesting system. It was a system composed of 10 peripheral processors from a CDC 7600 with a large shared central memory. However, the system was not a standard product, and a simulation showed that for our type of architecture the multi-processor system got in the way of responsiveness. The DataGeneral proposal met the hardware specifications, but the software could only address 96k of the 128k address space so it did not fit the software specifications. Edson DeCastro, President of DataGeneral, visited PROMIS Laboratory to discuss their proposal.

Varian Data Machines submitted the winning proposal with their V77-400 computer system (Fig. 9). It fit both the hardware and software specifications and the company was interested in working with us on the joint development of a communications subsystem to support the high-speed touch screen terminals.

FIGURE 9 The V77-400 in the temporary computer room. Photo by Jim Wanner.

TOUCH SCREEN TERMINAL PROCUREMENT

There were three departures from the mainstream in our terminal specification: the touch screen interface, the dual display modes (24 lines by 80 characters for selection and 36 lines by 100 characters for text retrieval), and the high-speed data transfer. After discussions with over 30 potential suppliers, we decided to drop the dual display modes as an essential requirement. We received eight responses to our specifications, and MEGADATA Corporation was selected to build the new high-speed, touch screen terminals. The terminal was a random-logic PDP 8 so that it could be programmed to perform our specialized functions. It had expansion slots for the two special boards to be built by MEGADATA, one for the touch screen and the other for the high-speed line interface. The touch screen did not have set pad locations on the screen, but was an x-y digitizer that had a resolution of 0.25 of an inch. It sensed a finger touch via echoing of surface waves. The first group of 20 terminals cost $8,500 each.

COMMUNICATION SYSTEM PROCUREMENT

The terminal-to-computer communications remained largely undefined until a computer system was selected in early March 1974. Soon thereafter a suggested communications system was formulated, using the standard Varian Data Communications Multiplexor and VTAM software as a starting point. A research contract was signed with MITRE Corporation in July 1974, to assist us in designing and implementing a communications system that would support digital traffic on a CATV-type distribution system for the PROMIS architecture. The resultant system was to provide highly reliable, low error rate and flexible reconfiguration capabilities for the PROMIS digital signals among the many terminals and several computer nodes. The data rate was to be at least 15,000 characters per second. Of secondary importance was the ability of the system to simultaneously support the communication of other services such as entertainment and educational television. MITRE was selected for this work because of their pioneering efforts with the MITRIX system, which used CATV technology to build a time-division multiplexing scheme on a CATV bus for inter-node data transfer. They had developed CATV modems as part of the MITRIX system.

PROGRAMMING LANGUAGE DEVELOPMENT

The only high-level language available on the V77-400 was FORTRAN. Its code was not re-entrant, and it had an extensive run-time environment that was not appropriate for our tasks. We had done extensive programming language research. I wanted a language with syntax that would support structured programming; a semantics that would support the manipulation of logical records of variable type and length; and a pragmatics that would support frames, records, and indexes to both. We wanted to be able to manipulate strings that could grow to be very long (up to 32k characters) and to be able to access mass-memory resident data logically. Facile control of the touch screen terminal was also a requirement, as was network access to the logical data structure and network access to multiple communicating processes.

I designed a programming language based upon our CDC 1700 assembly language experience and a study of other languages. Morgan and I wrote an interpreted version of the language on the CDC 1700 with semantics as defined above. The interpreted version accessed the data using a structure similar to a list processing languages' "property lists." Extensive searching through the data structures was required for any operand access, and consequently it executed very slowly. It would "run like the wind" and could take more than 30 seconds to process one selection.

We decided to develop a compiler language, the PROMIS Programming Language (PPL). PPL was a combination of a high-level procedural language with a very powerful embedded database

management system. It included procedures to manipulate the touch screen, schedule and sprout processes throughout a network, transfer data across the network, and manipulate strings effectively. It was efficient in terms of CPU cycles because it was not an interpreter; all data access to 16 bit numerics was in-line code and the internal form of the data required no searching. Each data element was accessed either with an absolute address or with a pointer variable and a relative address that was bound at run-time. Cantrill wrote the code generator, and I wrote the scanner and parser for the compiler. It compiled source code at 2000 to 6000 lines per minute.

The syntax of PPL was a major departure from "ALGOL-like" languages of that era. PPL's assignment operation was from left to right, the way one reads English. PPL had no explicit statement delimiters, that is, no ";" after each statement. PPL's comments could appear after implicit statement delimiters, and except for an * at the start of a new line, there were no explicit comment delimiters. Control statements all had explicit statement terminators. For example, an IF statement was terminated by an ENDIF or FI. Lists of statements could replace a single statement without requiring a BEGIN and END for the block. The syntax of a control statement required it to be spread across multiple lines, and no more than one statement could be put on a single source line. The PPL syntax supported our ideas of what "structured" code should look like.

PPL was designed for the programming of applications. We could not afford to wait until the compiler was done before we began programming its run-time support routines. We were also concerned about the efficiency of PPL for systems programming tasks; since PPL code was to be machine independent, we were also concerned about distorting PPL to fit various systems programming tasks. For these reasons, Cantrill developed a structured preprocessor to the Varian supplied macro-assembler. Called STRAP, it allowed all assembly language code to pass through. Statements recognized as STRAP commands caused output of the appropriate macro-assembler instructions. The preprocessor was originally written in STAGE2, a machine independent macro-processor, but it processed code too slowly. When written in STRAP itself, it could process up to 5000 lines of code per minute.

Software and Hardware Specification

Our funding agency was devoting a large percentage of their budget to our development efforts and wanted to be assured that we were up to the task. An advisory committee was put together to oversee our development. The committee included Ivan Sutherland, Alien Newell, George Robertson, Herbert Sherman, and Jack Hall, M.D. Newell and Robertson had previously developed a menu selection system called ZOG, which sensitized them to our work. They applied the PROMIS interface ideas of a rapid response, large network system to ZOG.

The implication of the new architecture involved coordinating efforts of multiple hardware vendors and the PROMIS Laboratory computer group. We had limited time and money. We followed D. L. Parnas's advice. The implementation used

> specifications sufficiently precise and complete that other pieces of software can be written to interact with the piece specified without additional information... The specification must be sufficiently formal that it can conceivably be machine tested for consistency [and] completeness [in the sense of defining the outcome of all possible uses]... By this requirement we intend to rule out all natural language specification. (Parnas 1972)

We had been using decision tables for program development for a few years. Henry Beitz, a CDC consultant, convinced us that an unlimited entry decision table could be used as the basis for demonstrating an understanding of a problem and also as the fundamental design document. Decision tables were used to design and test the interface among hardware and software modules.

The run-time environment developed to support PPL (described in the next section) was defined in 17 decision tables and 78 rules. The definition of the polled multi-drop protocol to connect the high-speed terminals to the CPU was a set of 12 decision tables with a total of 77 rules. The protocol was used by Varian Data Machines, by MEGADATA, and by PROMIS Laboratory. When the hardware was delivered and the terminals were connected, the system operated correctly the first time it was tried. One problem was discovered in an error pathway in the hardware, and six problems requiring software modification were discovered. We also used "thin-wire" protocols to define our run-time environment. We sent messages among three separate tasks: the PPL code environment to manage the resources of the CPU, terminal input/output to manage the terminal and printers, and block input/output to manage mass-memory.

Run-time Environment Supporting PPL

The PPL run-time environment consisted of facilities to support the work requirement modularized into processes, to support the internal form of the PPL logical data type (the paragraph) and to minimize the number of mass-memory accesses.

PROCESS SUPPORT

A process required access to three types of resources: The CPU for execution of PPL code, the mass-memory devices for access to the PPL paragraphs, and the terminals and printers. Each of these three types of resources was accessed using a task in the Varian Data Machines VORTEX II operating system. Based upon the resource required by the PPL code being executed, messages dispatched the PPL process among the different tasks.

SUPPORT FOR THE PPL LOGICAL DATA TYPE

Files and individual blocks were manipulated by the file system. The PPL logical data type was broken up into paragraphs, sentences, and sentence elements and were manipulated by sentence input/output routines. The file system, developed by Henry Stambler, used the VORTEX II software as a foundation, and provided the following enhancements:

1. A file was configured as a set of discontiguous components spanning several VORTEX file areas or disk packs. Up to 64 components were allowed for each file.
2. New components could be added as needed.
3. A file could be as large as 2048K blocks. In VORTEX the maximum was 32K sectors. A block could be up to 2K bytes long.
4. The table defining the files resided in central memory so that any block could be obtained in a single disk access.
5. Software check-summing and read-after-write validation were provided.

The sentence I/O routines, developed by Morgan, performed the mapping between the PPL operands and words within the file system blocks by maintaining state variables and a map-in area for each environment, making possible all the logical and string operations in PPL.

MASS-MEMORY ACCESS

PPL programs required rapid access to large amounts of data. It was impossible to keep even a small fraction of this data in central memory, and it was undesirable to access mass-memory every time a piece of data was needed. A memory buffer pool, developed by James LeMay, housed all data accessed via PPL instructions. The data in the pool ages, and because a PPL program normally accessed the same information many times during its execution, the pool frequently still contained the information. The pool was a resource used by the resource management tasks; it was transparent to the PPL programmer. The ratio of block to disk accesses was typically between 2:1 and 25:1, based upon the type of program and the amount of data it needed.

The Library System: Management of the Medical Knowledge

Medical knowledge entry was controlled in a manner similar to patient record data entry and retrieval. The library builder made selections at a touch screen terminal. By answering questions and selecting the next step in a series of actions, the builder was guided through the steps required to add new information to the database or update previously entered knowledge. The library system was developed to manage the frame and table library, in the same way that PROMIS was developed to manage the patient care functions. PPL was used to implement the library system on the same hardware base that the PROMIS system was implemented. The library system managed frames and tables.

TABLES

Tables were developed for many uses. The information in a procedure table was used, for example, to

1. determine branching to a subroutine of frames at certain points in the frame sequence;
2. define information that is specific to a site (such as drug prices and inventory levels);
3. define lists of logical actions that should take place when this procedure is stored in a patient's record;
4. specify lists of synonyms for the procedure to make it accessible on various alphabetic lists;
5. classify the procedure for all uses;
6. point to other data structures that are related to this procedure; and
7. list all frames and tables that access this table.

FRAMES

The frames were interrelated in a network; each choice on each frame pointed to a frame to be displayed when that choice was selected by a user. The network was a guidance system in which every frame was simultaneously medical content and structure for the next finer level of content. The most general frames, superstructure frames, served as high-level indices to other frames in the network, establishing the context (such as the current section of the medical record) the user would be working in. As the user progressed through the network, the frames became more specific, containing, for example, drug information or lists of diagnostic procedures, and enabling procedure ordering and reporting (see Fig. 10 for a frame sequence).

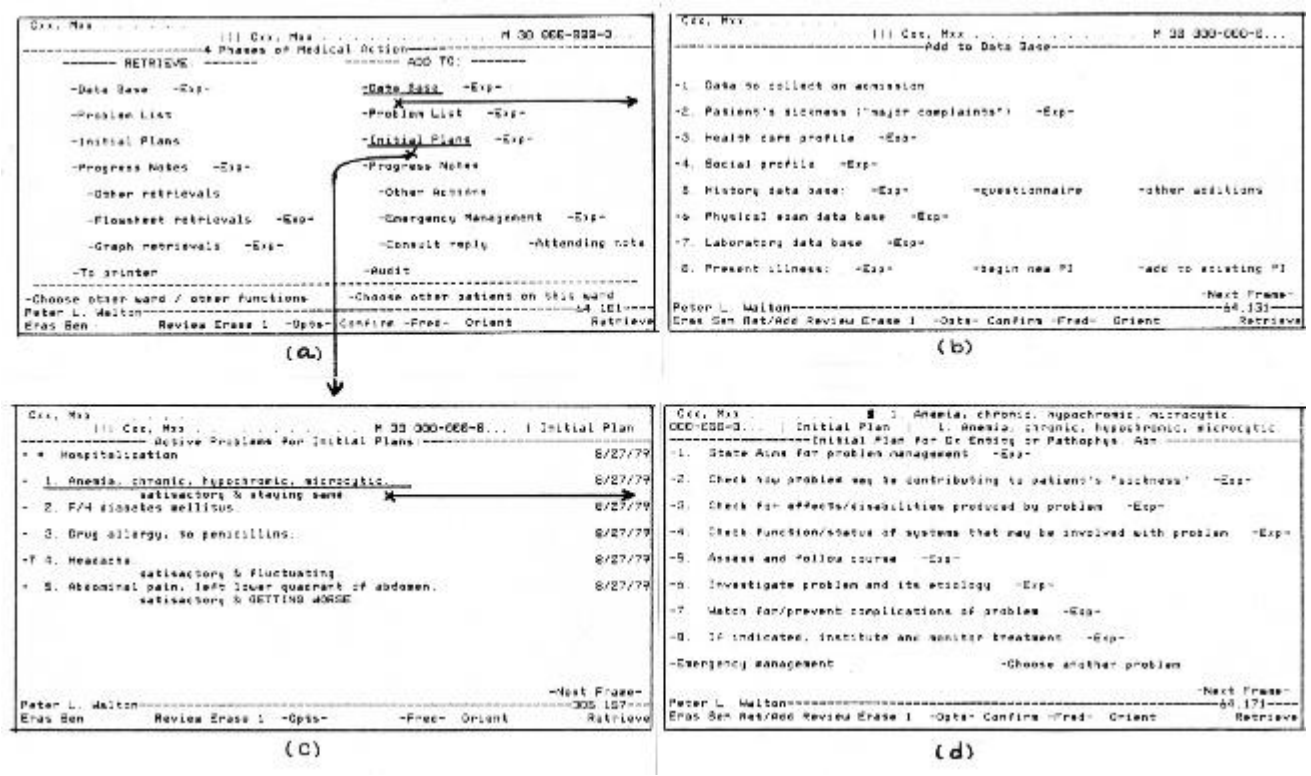

FIGURE 10: **Example of guidance frames:** Frame (a) is the initial frame that was presented after a patient was selected from the list of patients whose records the user was allowed to access. From this frame, all additions to and retrievals from a single patient record can be performed. (Note the titles RETRIEVE and ADD TO near the top of the frame.) Selecting Data Base branches to the add to Data Base frame (b). Selecting Initial Plans, on the add side of (a), brings up a list of the patient's active problems. All patient care actions are taken in the context of a specific problem. If Anemia were selected, as in the example, the appropriate planning structure frame (d) is presented. The displays contain frame content as well as context information for the user. The context changes as the user makes selections. The user's name is displayed in the lower left corner. Once a patient has been selected, the name is shown in the upper left-hand corner. The selected problem is on the top line in the center.

SIZE AND MAINTENANCE OF THE FRAME AND TABLE LIBRARY

By 1979 more than 10,000 tables and 37,000 frames comprised the library. These frames and tables contained information on about 2,700 medically defined problems, 640 drugs, 800 clinical laboratory procedures, 200 radiologic procedures, 700 diets, activities, physical therapies and ward procedures, a comprehensive physical examination, patient history questionnaire, and problem-oriented superstructure.

Content was built in frames and tables from accepted texts and from articles in medical journals that satisfied selection criteria. More than 60 journals were regularly reviewed by medical librarians. Source, date, and builder of each piece of content was recorded in a reference. Auditing of existing content was performed continuously.

The frame and table library was maintained via an interactive frame and table editor (FRED—originally for FRame EDiting only), a syntax checker, a pathfinder, and other tools. The syntax checker reported violations of the rules used to build the frames and tables. The pathfinder found all frames and tables that access a given frame or table. FRED manipulated the content in frames and tables. FRED was table driven; the type and form of each data element manipulated by FRED were specified in a set of tables accessed by FRED. FRED initiated back-pointing and indexing among the frames and tables.
The frames and tables referenced each other in order to define display branching or to use an entity. It was necessary to know how and where each paragraph was referenced to effectively manage the frame and table library. All back-pointing was performed by a routine scheduled by FRED.

CAPTURING MEDICAL KNOWLEDGE

The medical content form and structure evolved from emphasis on providing guidance to capturing the information needed to build the guidance capability. From 1976 to 1979 Drs. Peter Walton, Robert Holland, and Stuart Graves worked with the assistance of Larry Wolf, a software specialist, on the means of capturing and codifying the medical knowledge. What evolved was the idea of capturing the basic units of medical knowledge, the facts (Walton, Holland, and Wolf 1979). A fact can simply state a relationship among entities or qualify a relationship in greater detail:

> By stating relationships, facts provide a means for structuring medical knowledge. The structure can be viewed as a network of relationships. In qualifying relationships, facts capture the content of medical knowledge.
>
> Facts are the fundamental database of medical knowledge and are structured in ways that facilitate access to the database. For example, the fact in... [Fig. 11] is a "cause fact." Cause facts state an etiologic relationship between a *predecessor* (iron deficiency) and a *successor* (transferrin saturation) when the predecessor is considered sufficient to explain the successor (i.e. iron deficiency "causes" a transferrin saturation of less than 16 percent)... Predecessors and successors are used to define links which the system traverses to access the knowledge pertaining to a particular entity.
>
> The ingredients needed to offer computer-based healthcare guidance are now all in place. Knowledge is captured in atomic form in the facts. Facts are organized using a directed network. The network facilitates understanding and use of the facts to offer guidance. Building of this guidance capability is done in an explicit and rigorous fashion. Each step in the process of transforming knowledge to facts to guidance is documented and connected in electronic form to make both guidance and knowledge maintainable and adaptable.

```
Fact 135.00303
      Dx: iron deficiency
      Mf: transferrin saturation
      Value: below 16 % (average 7%)
      P[Mf|Dx] = 1

      Fact audited.
      Pr (coded): iron deficiency
      Su (coded): transferrin saturation

      Flags: p[Su|Pr] = 1;

      Refs:
        Clinical Hematology
        Wintrobe MM, et al., Eds.
        Philadelphia: Lea and Febiger, 7th ed., 1974
          Ch 16: Anemias characterized by deficient hemoglobin synthesis and
              impaired iron metabolism
          pp 621-634
      Process: 134.1  Dependence fact from authority source
      By: RR Holland MD-SP   1/17/79
      Documentation audited.

      Refs:
        Clinical Hematology
        Wintrobe MM, et al., Eds.
        Philadelphia: Lea and Febiger, 7th ed., 1974
          Ch 17: Iron deficiency and iron-deficiency anemia
          pp 635-670
            Laboratory findings
            pp 656-660
      Process: 134.1  Dependence fact from authority source
      By: GW Gilroy RPh-SP   1/22/79
      Documentation not audited.
```

FIGURE 11 Fact in expanded text form.

The Master Library and Instances

The library system incorporated ideas outlined in the Conway report on the National Library of Displays. There were three classes of information in the generic library:

1. generic information that applied to all instances;
2. local information that applied to only one instance; and
3. protocol information that may apply to several instances, but is designed and built with a specific set of values in mind.

Generic and protocol information were integrated into PROMIS by the library system. The generic library was responsible for maintaining all generic information and some protocol information. All other information was to be maintained locally by the instance that built and used it. All instance-specific data was identified in the PROMIS frame and table library, and procedures were implemented to allow the updating of only the generic information.

PROMIS Returns to a General Medical Ward

In December of 1976, the newly implemented PROMIS became operational on a 20 bed general medical ward (Brown III) in the Medical Center Hospital. The Brown III site was linked to the pharmacy, clinical laboratory, and the radiology departments.

Functionally the new PROMIS was a major improvement over the CDC system. It was easier to use and to maintain (Fig. 12). It had expanded functions and was more responsive and reliable (Fig. 13). Until September of 1981 when PROMIS Laboratory left the University of Vermont and became PROMIS Information Systems, Inc., there was a steady stream of improvements and new functions. Some of the features of the 1980 PROMIS system are described in the following material.

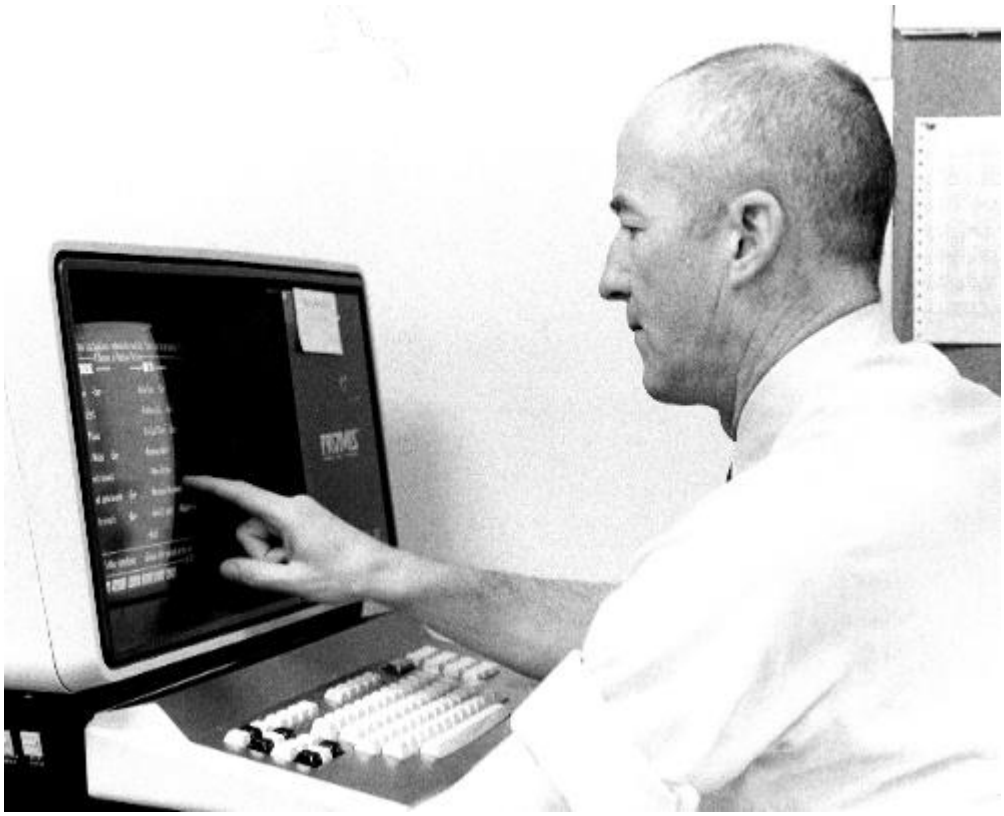

FIGURE 12 Larry Weed using Megadata Terminal.

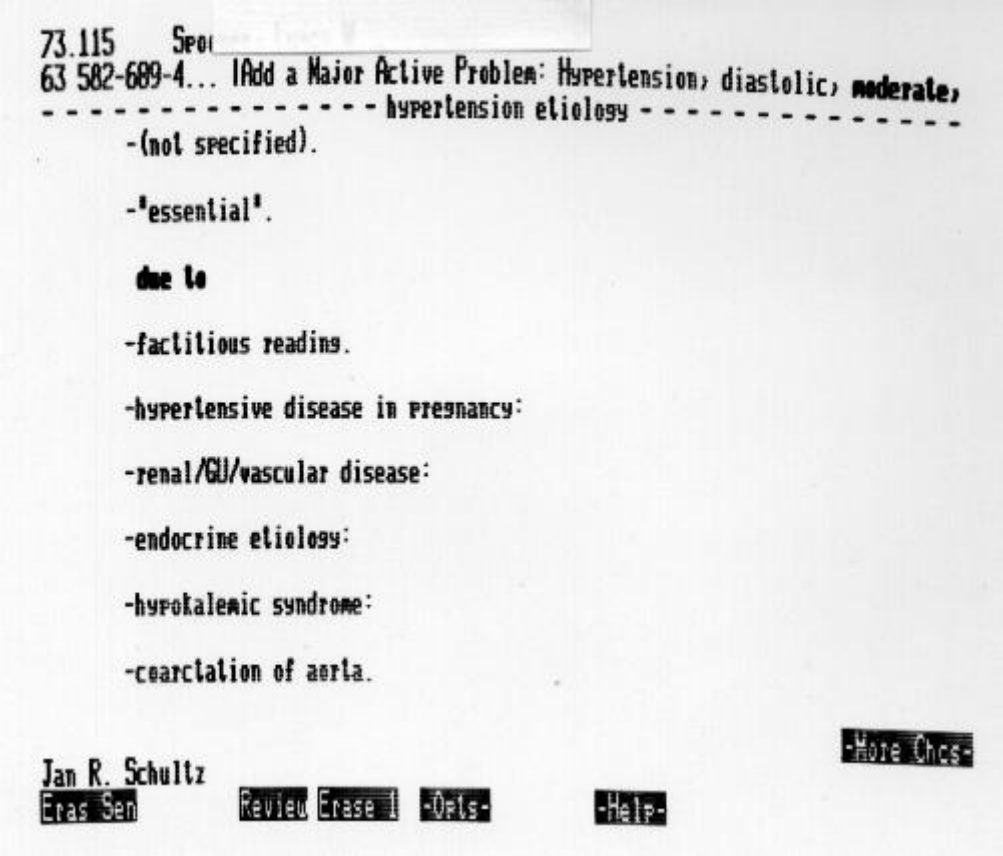

FIGURE 13 Megdata screen with inverse video function pads.

ADDITION OF ADMINISTRATIVE FUNCTIONS

The administrative functions that are the basis for most computer based medical systems were not initially integrated into PROMIS. LeMay, along with other members of the PROMIS computer group, rewrote an existing administrative package to operate in conjunction with the electronic patient record. The resulting administrative system included patient billing, accounts receivable, accounts payable, and general ledger functions. All manual input was entered at a terminal. Admission, discharge, transfer, and charge data were generated automatically as a byproduct of use of the electronic record. Input to the general ledger from patient billing, accounts receivable, and accounts payable was generated automatically.

NETWORKING WAS MADE ROUTINE

A two-node configuration was used routinely for PROMIS and system development; additionally, heavy simulated loads were applied to three nodes in order to validate the effects of contention for the basic network resources. Experiments were performed placing all terminals on one node and all medical records on another; this forced inter-node activity for each storage and retrieval operation. Under these conditions the user perceived little if any performance degradation due to the network.

POPULATION STUDIES WERE ADDED TO PROMIS

The ability to interrogate the electronic patient records as a group for patient care or clinical research was provided by the population study system, developed by Stephen Reynolds. Population studies attempted to answer questions a clinical investigator might ask of the patient records. The population study system enabled three types of interrogation:

1. Type of question: Who has a specified combination of attributes?
   Form of result: List of patients
   Example: Who is receiving gentamicin?
2. Type of question: What are the results of a set of specified procedures?
   Form of result: Array of values
   Example: What is the standard deviation of BUN in patient group Y?
3. Type of question: What problems appear on the problem lists of patients in a specified group?
   Form of result: Ordered list of problems
   Example: What is the incidence of all problems identified in patients in group Z who have diabetes?

Peer Standards Review Organization personnel at the hospital performed audits using the population study tools, the results of which were submitted to the hospital administration and medical board. Questions were similar to:

```
IF (date of serum creatinine) MINUS (date of digoxin) GREATER THAN 2
OR (date of digoxin) MINUS (date of serum creatinine) GREATER THAN 2
    THEN "Serum creatinine not ordered in time"
```

The result of this query would be a list of patients for whom the assessment "Serum creatinine not ordered in time" (as defined by the Boolean expression) was correct.

The processing and display of a patient set based upon the simple presence or absence of attributes was done while the user waited— usually within ten seconds or less; a patient set based on attributes qualified by time or other variables took longer depending on the complexity of the question and the number of patient records processed.

It was also possible to pass data from the electronic patient records to the MINITAB II statistical package (developed by the National Bureau of Standards and modified at the University of Pennsylvania) for statistical analysis. MINITAB was implemented with the commands as choices on frames, allowing easy, rapid, controlled, and structured access.

All *patient attributes* (see, for example. Fig. 14) within the electronic records that were used for population studies had to be coded. Complex relationships within one procedure report could not be captured. A coding mechanism was needed to parallel the text string seen by the user. The development of the code string, a tree-structured, frame-driven encoding mechanism, would have to wait for the next application design iteration.

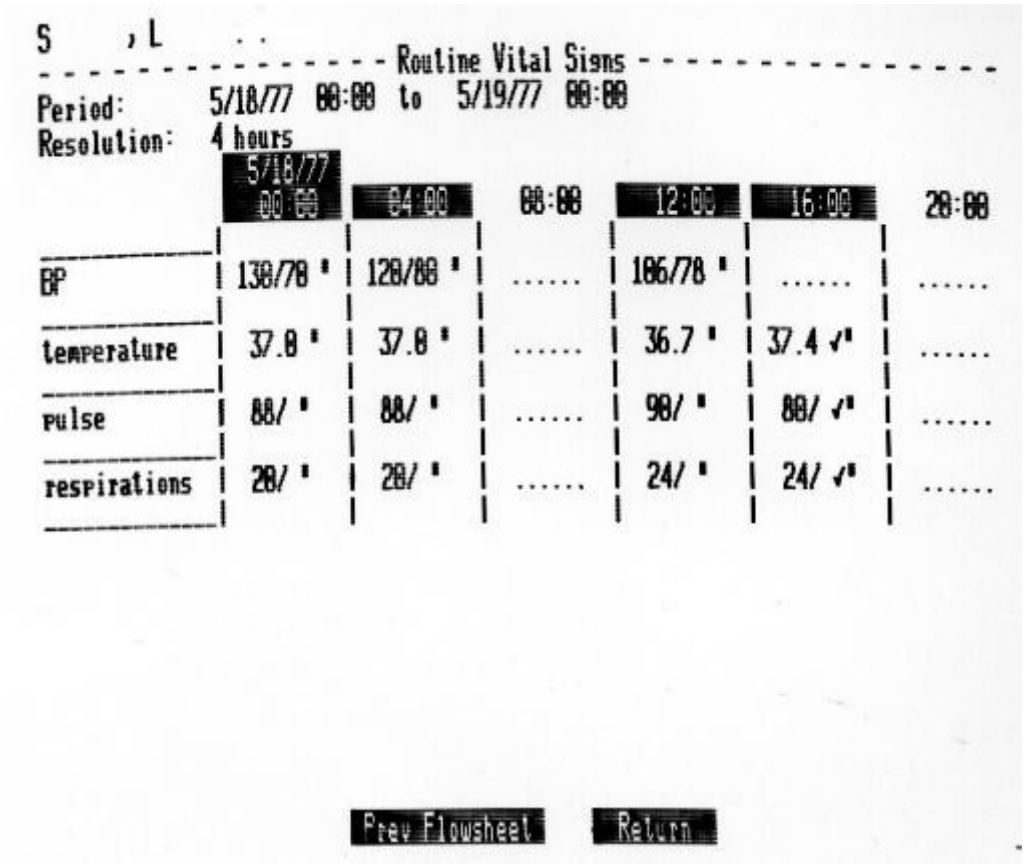

FIGURE 14 Example of vital signs flowsheet for a single day.

PATIENT LIFETIME RECORDS WERE DEVELOPED

The structure of the electronic record was expanded to allow multiple inpatient admissions and outpatient visits within one record. The *encounter* was the structuring mechanism for the patient's *lifetime record*. Each inpatient admission or outpatient visit was a new encounter. The user could specify any encounter (or all encounters) from which to retrieve patient data.

COMMUNICATIONS SYSTEM USE

The MEGADATA touch screen terminals, as well as printers and other peripherals, were connected to the CPU with a high-speed communications bus. The data communications lines could be extended up to 20 miles (32 km) by cable and further extended with microwave links. A very high frequency (VHP) 2-way coaxial bus was employed to provide many inexpensive interconnect ports. Full-duplex modems operated at 307,200 bits per second. Programmable controllers managed peripheral devices and operated at the remote interconnect ports. The line control protocol was a simple polling protocol. Each remote device was assigned a unique poll address. Because of the line protocol and the nature of the bus, users were free to move terminals and other peripherals from one location to another and continue without any access (sign-on) requirements (Wanner 1978). The bus was extended with a microwave link to the Grand Isle Medical Clinic (Fig. 15) 17 miles from the Hospital.

FIGURE 15 The Grand Isle Clinic. Find the microwave dish?

ALTERNATE POINTING DEVICE INVESTIGATED

From the early 1970s on, we received questions about the utility of the touch screen as our primary input device. We decided to investigate other commercially available pointing devices as part of the new PROMIS technology. Our investigations revealed that the only other commercially available pointing device was the light pen. We purchased an 8080 based terminal from Zentec, Inc. It accepted a light pen as a peripheral. Jim Wanner developed a special high-speed interface board to allow the terminal to connect to the CATV bus. He programmed the Zentec terminal for the CATV bus protocol and for emulation of a Megadata terminal with a light pen instead of a touch screen. The light pen was cylindrically shaped with a button on the side. A cursor would appear on the CRT screen if the light pen were pointed at a selection. When the button was pressed, the selection was made.

The Zentec light pen terminal was available in PROMIS Laboratory from April of 1976 until September of 1981. It was used by most computer and medical personnel associated with PROMIS Laboratory. On a number of occasions we presented it to our medical users outside of PROMIS Laboratory. There was a general consensus that it was not as usable as the touch screen. There were a number of problems with its use: First, the light pen had to be picked up, thus interfering with either typing or writing. Second, when the button on the side of the light pen was pressed to make a selection, often the pen's aim was disturbed, resulting in wrong selections.

Conclusion

HUMAN INTERFACE

The combination of a response rate of less than 0.25 of a second and the large database of frames allows users to navigate through the network of frames very rapidly, accomplishing their tasks without using the keyboard. The interaction rate is very high since a single selection results in the presentation of a full screen of new text. A complete thought can be communicated in a single touch. The database available to a user is a combination of electronic records generated from past selections and of frame and table knowledge. The user can operate from a universe larger than what can be remembered and has only to recognize the correct selection (and not recall it). These features provide for a unique and very effective human-computer interface.

The building of the database of frame and table knowledge has been time-consuming and intellectually challenging. The PROMIS frame and table library took over 200 person-years of effort to build. This type of interface can be successful if the cost of developing and maintaining the knowledge base used by the interface can be spread across many users.

In order for such an interface to work, both an extensive knowledge base and a different relationship to the computing resources it uses are required. For PROMIS, the interface is the computation and not a means to get to a computation. As computing power gets less expensive, this interface becomes more cost effective.

Robertson, Newell, and Ramakrishna discuss the PROMIS/ZOG type of interface and claim that this

> type of interface is a preferred mode of man-machine interaction, even over the use of natural language dialogue with the computer in the role of intelligent agent.
>
> There are two polar views about how to structure man-machine interaction. One is the computer as tool... Control remains with the user. The other view is the computer as intelligent assistant. In this view one wishes to make the computer more intelligent and communication with it more natural... Precisely what an intelligent assistant is supposed to provide is freedom (of the user) from the effort of understanding. Put one other way, delegation requires an act of faith.
>
> ZOG is an evolution in the tool direction. It seeks to produce a transparent device which, in itself, has no intelligence at all, but is immensely responsive to the user. It seeks to do this in the arena where we normally expect to use natural language, namely, dealing with large bodies of knowledge. Indeed ZOG uses natural language for its output (though arranged in a sort of spatial dialogue), for the user has good devices for assimilating it. However, its own internal structure, which governs what it says and when it chooses to say it, is completely open to examination by the user. (Robertson 1977)

APPLICATION DEVELOPMENT

In October of 1981, PROMIS Laboratory left the University of Vermont and formed PROMIS Information Systems, Inc. We realized that the potential for the PROMIS technology was broader than medicine alone and that work was needed on the application development software. From the very beginning of our development in 1967, the Human Interface Program interpreted the frame in the process of bringing up the next display. It was table-driven since the frame (the programs table) defined how the program was to operate. Other parts of the system were not table-driven. The frame editor has been table-driven since 1978, but the storage and retrieval programs had embedded within

them the logic of the problem-oriented medical record and could handle only one type of record structure. We have rewritten the application subsystems so that they are all table-driven. This allows the development of systems as large and complex as the PROMIS medical application without writing PPL programs.

Application development without writing programs makes programmers more efficient by a factor of between 10 and 100. What we have developed is not a programmer-less system, but a set of tools that makes programmers more effective. The tool kit uses the concepts of programming to accomplish the tasks of programming, but the writing of procedural programs is not required.

CONTROL OF INFORMATION

The PROMIS development coupled medical knowledge with electronic medical records and increased access to both for all personnel who used PROMIS. The medical knowledge captured in frames and tables was available to all users. Since PROMIS was the medium through which users performed many medical tasks, the presence of the medical knowledge at the time they performed their work facilitated excellent performance without the reliance on an encyclopedic memory.

The electronic medical records were available to all properly identified health care personnel. The boundaries of space, availability of only a single copy, and legibility were gone. Users could access a patient's medical record from anywhere a terminal was located. Once accessed, the information was legible and presented in either narrative or flowsheet form.

Electronic medical knowledge and electronic records moved the control of the information from the physicians to other users. In other words, PROMIS shifted the source of power away from physicians. Examples of the shift for nursing and supporting areas (pharmacy, radiology) follow.

The role of nursing was enhanced by PROMIS in a number of ways: First, all orders written for a patient were immediately available. Second, all information added to a patient's record could be reviewed at any terminal; there was no queuing behind the single copy of the paper record. Third, medical knowledge was available at the terminal, and not as in the past only in textbooks or the user's memory. Fourth, the patient-administered questionnaire resulted in a complete medical history, which nursing used to help define the patient's problem list. At times this resulted in physicians demanding of the nurses: "Why did you give my patient all those problems?"

The role of the supporting areas (e.g., pharmacy and radiology) was enhanced. Each supporting area had access to the patient's record along with the requisition for service. The order could be checked for appropriateness using the full record to provide a context. New reports could be compared with older ones to see if significant changes had occurred. Results reported using the terminal would be immediately available on the ward. For narrative reports, there was no time lag because there is no need to type the dictated report.

THE FUTURE

When we began our work, a touch screen terminal operating over telephone lines at 2400 bits per second cost over $20,000. Today, a similar terminal can be purchased off-the-shelf for under $3000. The personal computer I'm using to write this paper has more central memory in it than the CDC 1700 system used for the initial development and the personal computer's central memory cost 200 times less. The technology is available now to make the type of interface and systems described in this paper cost effective and widely available.

The PROMIS system has been installed at the Baycrest Geriatric Hospital in Toronto, Canada. It is still the most advanced medical information system in existence and the only one to manage a fully

electronic medical record. It has yet to be used as the information system for a total hospital, but its time is coming.

ACKNOWLEDGMENTS

This paper represents the efforts of many individuals in medicine and in computer science, both within and outside the past PROMIS Laboratory. I would like to acknowledge the group from PROMIS Laboratory who have stayed together to keep these ideas alive. This work was funded primarily by the National Center for Health Services Research, Department of Health, Education, and Welfare. I would like to thank Adele Goldberg, Tony Stern, and Sue Burton for suggested additions and editorial comments.

REFERENCES

(Hertzberg, Schultz, Wanner 1980) Hertzberg, R. Y., Schultz, J. R., Wanner, J. F., "The PROMIS Network," *Computer Networks* 4, 1980, pp. 215-228.

(Parnas 1972) Parnas, D. L., "A Technique for Software Module Specification with Examples," *Comm. ACM* 15(5):330-336 (May 1972).

PROMIS Laboratory Staff, "The Representation of Medical Knowledge," *Proc. Second Annual Symposium on Computer Applications in Medical Care*, Washington, D.C., November 1978, pp. 368-400.

(Robertson 1977) Robertson, G., Newell. A., Ramakrishna, K., ZOG; *A Man-Machine Communication Philosophy*, Department of Computer Science, Carnegie-Mellon University, August 4, 1977.

(Schultz, Davis 1979) Schultz, J. R., Davis, L., "The Technology of PROMIS," *Proc. of IEEE* 67:9, September 1979.

(Schultz, Cantrill, Morgan 1971) Schultz, J. R., Cantrill, S. V., Morgan, K. G., "An Initial Operational Problem Oriented Medical Record System—For Storage, Manipulation and Retrieval of Medical Data," *AFIPS—Conference Proceedings*, Vol. 38, 1971.

(Walton, Holland, Wolf 1979) Walton, P. L., Holland, R. R., Wolf, L. L., "Medical Guidance and PROMIS," *IEEE, Computer*, November 1979, pp. 19-27.

(Wanner 1978) Wanner, J. F., "Wideband Communication System Improves Response Time," *Computer Design,* December 1978, pp. 85-91.

(Weed 1964) Weed, Lawrence L., M.D., "Medical Records, Patient Care and Medical Education," *Ir. J. Med. Sc.* June 1964, pp. 271-282.

(Weed 1968) Weed, Lawrence L...M.D., "Medical Records that Guide and Teach," *New England Journal of Medicine*, 278:593-600, 652-657 (March 14, 21), 1968.

(Weed 1969) Weed, Lawrence L., M.D., *Medical Records, Medical Education and Patient Care*, Case Western Reserve Press, Cleveland, Ohio, 1969.

(Weed 1972) Weed, Lawrence L., M.D., *Problem Oriented System*, chap. 29. "Background Paper for Concept of National Library Displays," J. Willis Hurst and H. Kenneth Walker (eds), Medcom Press, 1972.

Participants Discussion

Allen Newell:
One of the things we got the feeling for, both in Jan's talk and in the one that House gave about Hewlett-Packard, is the notion of quite separate developments. But I wanted to at least put on record some rather interesting connections in which, in the technological society we deal with, things get tied together. A key person here is Bruce Waxman, who was a man associated with the Life Sciences effort at the NIH. He used to come around and bug me about how they've got this big AI program that's got 32,000 frames on it up in Vermont some place, and how I really ought to go and see it because they've sort of done what you guys are trying to do. I ignored him for the first couple of years. Finally, I went up there and saw the PROMIS system and, with a colleague of mine, George Robertson, ended up on a technical committee to review some of their work for a couple of years. We ended up building a system called ZOG at Carnegie-Mellon, which was a direct follow-on and copy of PROMIS. So, even in a conference like this, there are connections that seem really quite separate. In fact, the influences have moved back and forth.

John Sechrest:
I'm curious about how you see the PARC user interface affecting the interface that you have for PROMIS. I'm also curious how you got an overview of a patient's data; not how you looked down a road, but how you looked at the road map from an overview point of view.

Schultz:
The interface we used has been around a lot longer than the PARC interfaces. Ours is a talking narrative interface; pictures and graphics could certainly change it. I would imagine that over time the two approaches will merge and we'll have a slightly different sort of interface.

As to the second question, there are many different ways you can look at a patient's record. There is no single overview. Depending upon who you are and what your interests are, you want to basically look at it in different ways; our tool facilitated multiple views. There are problem-oriented records, and source-oriented records; you could retrieve all of the data, all the laboratory data, all the radiology data. There are time-oriented records; you could just retrieve the data chronologically; you could retrieve it by problem. Basically, you could take any view you wanted of this data, unlike most other record systems. We had placed eight different indexes on a single record, and you could traverse those indexes in different ways.

J. C. R. Licklider
I was on one occasion one of the visitors, and I will attest that the taxi driver did indeed know how to get there. I was very much impressed with the rapid touch screen action. I believe it was true that most of the people in the hospital who used the system liked it very much. They all gave it very high marks. On the other hand, I noticed that in the marketplace such screens don't seem to be going very well. My impression is that Hewlett-Packard ceased to emphasize touch screens as a selling point. Maybe that's wrong, but what's your perception of the touch screen as a contender in this mouse/light pen sweepstakes?

Schultz:
You can't look at just the touch screen. Our company is now supporting all pointing devices. If someone wants to use a mouse, they can use a mouse. As long as the system is responsive, you don't need a "gorilla arm" to use a touch screen; a light touch will do. My right arm is the same size as my left one, and I've been using a touch screen for over 20 years. You can't just say touch screens, light pens, mice— you have to say, "What is the whole system?" I think ultimately people are going to want

to use a touch screen more than a mouse; what's more natural than pointing? The parent has to tell the kid, "don't point." Well, we all want to point.

I used to go up on the wards. We had probably five thousand patients take their own history using the system. We would just sit them down and say, "Read what's on the screen and point at the thing, you know, just touch the yes or no." The patients would do it. And after a while, I would ask, "Well, do you realize what you were doing? In fact, it's branching when you touched the screen." And they said, "No, I'm just answering the questions." That's the kind of interface that you want. The computer is completely transparent, and what becomes important is the knowledge behind it.