Integration of Computer Support for Institutional Practice: The HELP System

Homer R. Warner, M.D., Ph.D., T. Allan Pryor, Ph.D., Stephen Clark, M.S. and John Morgan, Ph.D.

The purpose of this paper is to report on recent efforts in using computers to facilitate the practice of medicine in the hospital environment. The work to be presented is the work of a number of people in our department and constitutes, to this point of development, something over 200 man-years of effort. We shall explain some of the philosophy of the system, describe a particular application and then give some of the results of that application's impact on the practice of medicine in a particular hospital. A decision was made some time ago that we would focus our efforts primarily in the area of medical decision making.

There are about eight assumptions that go into the design of the kind of system to be described here. Since the primary information-processing task in medicine consists of mapping the intersect of the body of medical knowledge onto the data obtained from a patient in order to manage his illness, this type of decision-making activity will be the most fruitful to pursue in search of our specified goal, which is to apply computers to effectively improve the practice of medicine. All other components of the system we design will be implemented to serve this particular function, namely, decision making. Second, since most decisions are based on data from several sources, an integrated computer-readable record on each patient will be essential. This is in contrast to some of the other kinds of decision-making efforts that have been made with computers in medicine, which, in general, require the input of the data upon which the computer will operate at the time the decision is made; in this regard, these systems suffer from the limiation that the physician must input the data right at

---

Homer R. Warner, M.D., Ph.D., T. Allan Pryor, Ph.D., Stephen Clark, M.S. and **John Morgan, Ph.D.**, Department of Medical Biophysics and Computing, University of Utah College of Medicine, Salt Lake City.

the site. The bulk of the data entry effort is his, rather than having this supplied to him from other sources. The third point is that most of the data input must not rely on either the MD or the nurse as its source. Fourth, some feedback must be given at the site of data entry for motivation and quality control. That is nothing we have invented; these are things we have decided are important to take into consideration in designing a medical decision-making system. There is no question about the kind of quality control that is possible if information can be fed back right at the site of data gathering.

An example of this is our ECG monitoring program, which is done completely on-line, from the television set in the patient's room. The technician simply plugs into the phone line and then gets feedback through that television set over an otherwise unused channel that allows her to get the quality control by display of wave forms and the diagnostic statement in semicoded form right at the site, without disturbing the patient. The test can be repeated if it does not agree with some criteria that she has already learned.

The fifth item under these assumptions is that the system must know when to consider a particular decision. Most errors in medicine occur through failure to even consider the right decision. We call this system "data-driven"; that is, it is not necessary to ask for a particular decision to be considered; it comes up automatically as a result of the data one has just stored in that patient's file. Sixth, parallel manual operation will result in neglect of the automated system. The user must rely upon the system if it is to be of any value. He cannot have a parallel system, or this becomes simply an experiment. We learned that in the early days of patient monitoring. Seventh, one or more key individuals from the subject matter area of each subproject must be involved to the extent that it becomes his project to sell to his colleagues.

Using ECG as an example again, it was not until we got one of our top cardiologists sufficiently involved in the project that he became enthused that he was able to sell it to the other cardiologists in the hospital. As long as the people who wrote the programs tried to sell it, because we were outsiders, it would not be accepted.

Eighth, the level of reliability expected of an automated system far exceeds that of its manual counterpart. People forget the unreliable features of the manual system and have almost absolute standards which they apply to the automated version. As a result, we have duplicated hardware components at every step. A computer-based decision-making system must be modular to be useful since it will never be complete. If you wait until you have a "total" system, you never implement anything. You implement one piece at a time, but you build it so that you do not have to re-do the first piece when you implement the second piece.

Finally, since the system must ultimately be controlled by the people it serves, interaction with it must be in natural language. This means that all input is in natural language and all output is in natural language. The people who control the medical logic must be able to do so in a form that is very simple and natural for them.

Figure 1 is a block diagram that shows very simply the set-up and how we have implemented this philosophy. The center block, labeled the CPU, communicates with a number of files. The patient data files (described in more detail below) are structured in order to optimize this decision-making process. The medical logic files are thus labeled because they are independent of programs. The medical logic is read in by a common program when it is appropriate. The same program that processes ECG patterns to classify them also makes decisions about the pharmacy, etc. We

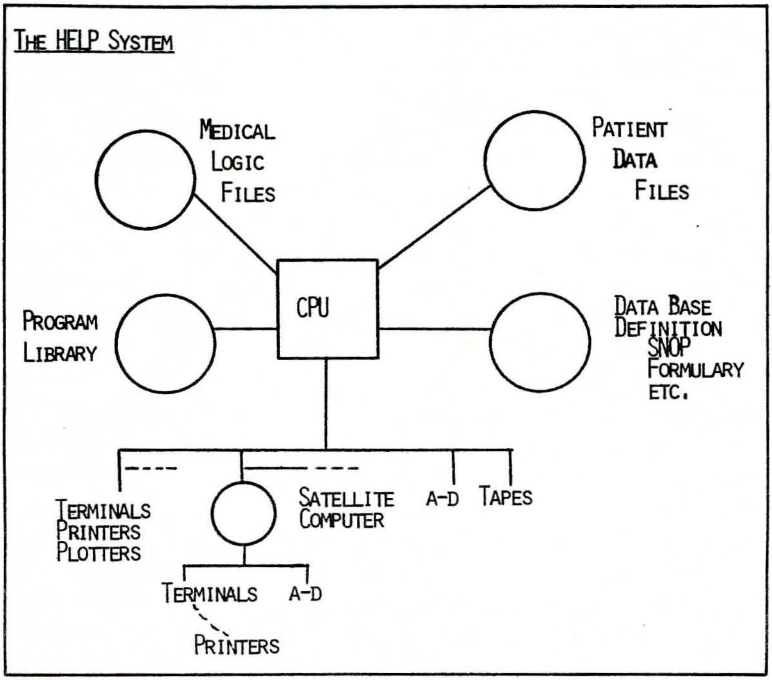

FIGURE 1.

do not write new programs each time we have a new medical logic area to operate in.

The program library is on another file, of course. Still another one is labeled data base definition. This is a very important feature because it permits the storage of the data in such a form that the English language or any other natural language can be used for data input and output, and still the data stored in a particular patient's file are all in tightly coded form. The definition of these codes is contained in this file. For instance, disease definition and even procedures, such as an operation, are all done with an expanded form of SNOP. The SNOP entry, again, has been formed in such a way that all interface with the SNOP is through English language. The SNOP coding is all automatic. The drug formulary is structured in such a way that it is appropriate for decision making. That is, the drugs are organized according to chemical structure so that there is a hierarchy; if a particular patient has an allergy to a particular class of drugs, that can easily be specified in the logic without having to specify in detail all the particular components of the class to which he is allergic.

Finally, there is a set of peripherals that actually controls the whole operation. We have no computer operators. The machine is run from remote terminals. We presently have over 70 terminals directly communicating with the central machine. These terminals are in a variety of forms: some are graphic terminals; some are typewriter keyboard input-type terminals; some are remote printers and even a few plotters. We also have four minicomputers, satellite computers, in the system. Most of our expansion will take the form of additional networking of small computers and eventually we expect that the small computers will take over the whole job. This relieves the load on the big computer, particularly for things like ICU monitoring, where there is a good deal of front-end processing that can bog down the big machine. The clinical laboratory is on a remote computer. All that automation is done ahead of time and simply passes the data to the patient file in the central machine.

The structure of the patient file is diagrammed in Figure 2. First, each patient has an identification file that contains some time-invariant information (the ID sector). Also in this file is a set of pointers to the other classes of data that the patient may have. If a particular slot reserved for pointing to the ECG file has a zero in it, this means that the patient does not have any of that kind of data yet. Likewise for chemistry or whatever. But if it contains a nonzero value, that will be a pointer to location of that particular class of files. Within each data class, the fields are variable format and variable field length. In general, each one is

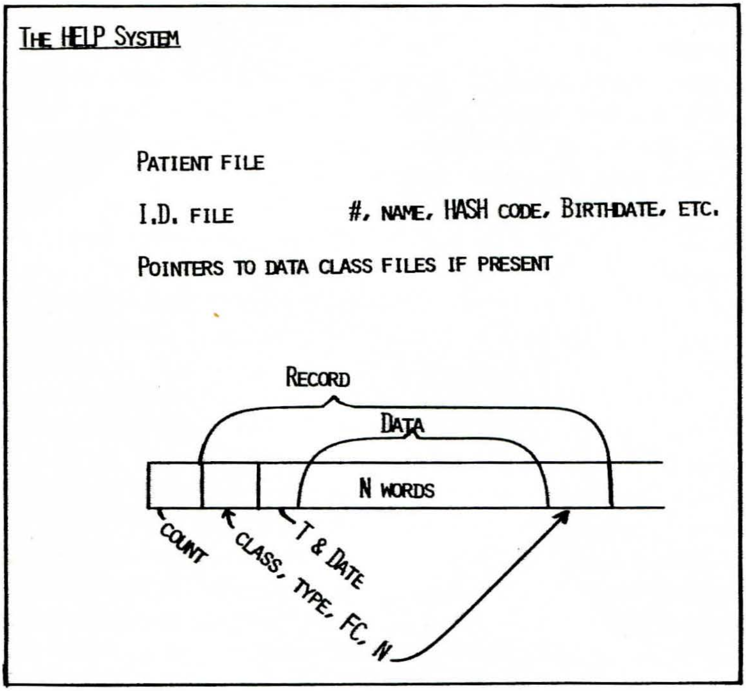

FIGURE 2.

time-labeled so that we have the time and date that the data were collected, not necessarily when they were stored. For instance, a chemical value will have the time at which the blood sample was drawn, not when the test was finally stored in the patient's record.

Table 1 shows some of the data classes that we presently have. For instance, all the different kinds of histories and the physical exam data are in one class. Comments and nurses' notes, particularly from the ICU area, are in another class. Medications and all of the formulary, as well as diet information, are in another class. Various kinds of laboratory data, including physiological as well as the clinical laboratories and the SNOP codes, are in separate classes. Number 12 in the table is called HELP decisions. As a decision is made by HELP, that decision is stored in another data class, as another piece of information, and the decisions can then be used as data for making other decisions. Thus HELP is an hierarchial decision-making system. Finally, the patient management protocol is another class of data that is just getting started in the outpatient area.

Table 1. The HELP System: Data Classes

1.	History and physical examination
2.	Comments, notes
3.	Medications and diet
4.	Chemistry
5.	Hematology
6.	Bacteriology
7.	Blood gas
8.	ECG
9.	Cath lab
10.	Pulmonary function
11.	SNOP
12.	HELP decisions
13.	Patient management protocols

The formats of these data have evolved over a period of years (Table 2). First there is what we call type 0 data. This is the kind of data format that one would use when each item is defined by its starting bit and the number of bits in a particular field. For instance, in the SMA-12 where you always store 12 values, there is a fixed array and the physician in that field can define what the data item is. Another example of type 0 format would be history data, where you simply want to record, say, two bits for each question, one that the question was asked, and the second whether the answer was yes or no. With type 1 data format, an item of information

Table 2. The HELP System: Data Coding Structure

Type 0	Each item is defined by its starting bit and number of bits in a particular field. (Example – SMA-12)
Type 1	Item defined by a string of 6 bit codes made up of a noun plus 1 (or more) adjective and adverb, class modifier or field code modifier. (Example – drug formulary)
Type 2	SNOP codes with modifiers which identify whether entry is admitting diagnosis, surgical diagnosis, pathological diagnosis, etc.
Type 3	Item is a decision made by HELP based on other data, where the code for the decision includes the data class and field code of the primary data used in that decision. (Example – ECG pattern classification)

is defined by a string of six bit codes, made up of a noun with one or more adjectives, an adverb and class modifier or field code modifier. This gives a very flexible structure for something like a drug formulary or the physical examination, where there may be a very broad set of possible data items to be stored in a particular patient record; but in any one particular case, you store a very small subset of that. So in this case you need flexibility to simply define the specifics of the item stored without having a fixed set as would be required for the type 0 data.

Type 2 is reserved for the SNOP codes. We have in the neighborhood of 40,000 terms now in SNOP. SNOP is really a nomenclature, not simply a fixed set of disease or code combinations. Any number of codes from any one of the four SNOP fields can be used to describe the particular term in which you are interested. We have now included a fifth field, called the treatment field, which includes all of the operation codes and gives us a very flexible system for coding information that is input by the physician.

Reference has already been made to type 3 data. This format is for decisions that have been made by the HELP logic itself. These decisions are stored in the computer and represent the second or higher order data in the system, "deduced data."

The system for data base definition is an on-line interactive system that allows a particular user, once he has been assigned a class of data, to control the data class and to define the data items in that class from a terminal (Table 3). He must define a name for a data item, either an English language name or a string of terms to describe it. He then can define key words for accessing the data, which can include words in the names or synonyms, to provide a convenient way of getting it from just English language input. He also has to define the format to be used and

Table 3. The HELP System: PTXT

A program for data base definition including:

1. Name of data item
2. Key words for later access
3. Definition of type and codes within type
4. Enter routine identifying data item by entering key words
5. Translation algorithm for retrieving text from data codes
6. File maintenance routines

then the system automatically assigns the codes for him, to prevent duplication of the same code for more than one item. There is an ENTER routine for identifying a data item by entering one or more key words. The system then displays the items which contain all these key words and the user chooses from this list. There is also a translation algorithm for retrieving the English text from the data codes, the inverse function of ENTER. Finally, there is a set of file maintenance routines for seeing that the data definition files are backed up appropriately and cleaned up when necessary.

Figure 3 is an example of a decision criterion from the HELP logic field and illustrates the way these are written in the system. This happens

```
THE HELP SYSTEM

        EXAMPLE OF HELP DECISION

        GENTAMICIN DOSE MAY BE EXCESSIVE FOR PATIENT'S RENAL STATUS.

            FVAL          (G * 10)    GT   H

        A     ANTI-INFECTIVES, AMINOGLYCOSIDES, GENTAMICIN, CURRENT, DOSE
                    LAST   FROM 2 MON BEFORE NOW TO 1 DAY BEFORE NOW

        B     ANTI-INFECTIVES, AMINOGLYCOSIDES, GENTAMICIN, INTERVAL
                    BETWEEN DOSES (HRS)   LAST

        C (18)  CALCULATED CREATININE CLEARANCE

        D     WEIGHT KG (x100)   LAST

        E  EX  (NOT (A AND B AND C))  EXIT

        F     D/100

        G     (A * 2400) / (B * F)

        H     C * 7 + 200
```

FIGURE 3.

to be a decision that comes out with the message, "gentamicin dose may be excessive for patient's renal status." The FV is the final evaluation statement. Items A, B, C, etc. represent one of two things: they are either commands for searches to be performed on the medical record or they are logical arithmetic or probabilistic operations to be performed on preceding items. The first one says, "anti-infectives, aminoglycosides, gentamicin." This is working its way down the tree of the formulary. We could have stopped with just "anti-infectives" if we only wanted the whole class, or we could go down to a detailed particular drug, such as gentamicin. The modifiers on it happen to be field code modifiers. It says, do you want to know first "is it current?" and second "is there a code in there for dose?" In other words, the computer is searching for that drug and if it finds it in the record, will store the dose under item A. That will be the retrieval initiated by this statement. The dose of gentamicin will then be in item A as a numeric value. Now it restricts the search to search only for the last value from two months before now to one day before now, as here specified; so it looks at the current time and calculates the time interval for the search. If it finds the item, it will pick the last item in that time interval, and it will bring up the dose of that item.

Item B looks for the same drug, but it looks for the interval between doses, another numeric value in that data field. A and B can be used in subsequent calculations. For instance, under item E, there is what we call an EXIST statement. EX means exist. It will now examine that to see if the following is true. It says, "NOT A and B and C"; that is, if items A, B and C were not found, then it is a command to exit from this decision. In other words, if HELP cannot perform the rest of the operation, it just jumps out. It does not say that it is not true; it just says that no decision can be made.

It can also perform arithmetic operations, etc., but the final evaluation takes the result of calculation of item B – which is essentially the drug level to be expected based on body weight, frequency, dose, etc. – and compares it to item C, which is the creatinine level. Under item C, 18 refers to sector 18 under this particular class of sectors. Using sector 18, HELP has calculated the creatinine clearance. The creatinine clearance is generally not measured on patients, so it will use the creatinine level and go through some manipulations on that to normalize it and will bring that up so it is in appropriate form for use in this decision, which is: "Is this particular prescription for this patient going to result in a blood level which exceeds the level that would produce autotoxicity and maybe deafness?" That will be a function of the weight of the patient, the dose, the renal function of the patient (how fast he can get rid of it), etc.

The HELP system consists of a number of program (Fig. 4). The first one is HCOM, the HELP compiler. This HELP compiler not only is the mechanism by which a user gets into the system to create medical logic and store it on disc, but it also reduces that logic from the English language form in which the user enters it to a compact logical form for rapid execution and at the same time creates pointers from the data items in that decision to other decisions that use the same data. This means that when a data item is stored which is used in making a decision about gentamicin, it automatically will bring up that decision and any others that make use of that data item, i.e. "data-driven."

DATAUP is the program that every program calls to store data in a patient's file. This then uses the pointer system to bring up the appropriate decision logic. When a decision is made, that decision is stored in the patient's decision list as a new item of information and the pointer file is referenced again. This allows all decisions to be used for other decisions. Appropriate decisions will appear on reports from various laboratories, will produce alarms in the ICU, and will also provide a way of presenting this information in hierarchial form for the physician to review. It is very important that the information get back to the physician in the form that

THE HELP SYSTEM

DECISION SYSTEM IS DATA DRIVEN

1)	HCOM	CREATES POINTERS FROM DATA TO DECISIONS THAT USE THAT DATA ITEM.
2)	DATAUP	WHEN A NEW DATA ITEM IS STORED IN A PATIENT'S FILE, THE DECISION SECTORS WHICH USE THAT DATA ARE PROCESSED AUTOMATICALLY.
3)	DECISION LIST	WHEN A NEW DECISION IS ADDED TO A PATIENT'S DECISION LIST, THIS IS TREATED AS A NEW ITEM OF DATA AND STEP 2 IS REPEATED.

FIGURE 4.

he wants it. So we give it to him in this hierarchial form, with the highest order information listed first. This starts with the set of decisions that have already been made on the patient. Next he might request the reasons why one of these decisions was made and what raw data were used. But he may not even want to look at that. If he does, however, it can be provided in a reverse order down the decision hierarchy.

Figure 5 shows the results of the use of this system in one particular application, namely, the pharmacy. On January 1, 1975, we implemented the pharmacy system on one general medical ward and one general surgical ward. Every prescription coming from those wards to the pharmacy was entered by the pharmacist through a terminal to the patient's computer-based medical record. If an alarm came up as a result of a decision made from that prescription, the pharmacist called the ward to notify the physician, who then could take the appropriate action.

During the eight-month test period, we had from 75 to 125 sectors or decisions available. The reason this number is variable, of course, is that we were adding new decision logic as we went along. Since it is a modular

THE HELP SYSTEM

8 MONTHS' EXPERIENCE WITH PHARMACY SYSTEM

1) 1 GENERAL MEDICAL AND 1 GENERAL SURGERY WARD

2) 75 TO 125 HELP SECTORS IN USE OVER THIS PERIOD

3) 1560 PATIENTS MONITORED

4) ALARMS ON 5.5% OF PATIENTS

5) 75% OF ALARMS RESULTED IN PHYSICIAN CHANGING THE PRESCRIPTION

6) NOW HOSPITAL WIDE (550 BEDS) AT COST OF 41¢ PER PATIENT PER DAY

FIGURE 5.

system, you can add more and more. Some 1,560 patients were monitored on those two wards during that period. We had alarms on 5.5% of the patients — that is, not 5.5% of the drugs, but 5.5% of the patients. Seventy-five percent of those alarms were heeded by the physician in the sense that he made changes in the medications that the patients received. Perhaps most encouraging to us was that, at the end of that eight-month period of testing, the physicians wanted the system and we have implemented it hospital-wide. We now charge the patient 41 cents per day for the service, which is now an integral part of the patient care system.

This is one example of how we have implemented HELP in the hospital; it is one of the more striking examples. There are a number of other areas where we are presently using the HELP system, including the interpretation of all our blood-gas measurements. All our ECGs are analyzed with the HELP system, and patient screening is done this way. The sequential Bayesian history is implemented through the HELP system and provides diagnostic statements to the physicians. The clinical laboratory is partially implemented, with the electrolytes and the hematology. None of these applications ever becomes complete; they are ongoing, dynamic systems. In the clinical laboratory, we have a system that is still in the experimental mode, but it looks very encouraging for automated ordering and sequencing of laboratory tests, where the results of the first test determine the new test to be done and often these can be done on the same specimen.

Discussion

Question: Does the 41 cents per day include the amortization of the hardware or the development costs of the software?

Dr. Warner: It does not include any of the development costs. It includes the costs of operating the system and the prorated cost of replacement of the hardware.

Question: Does it cover the initial investment?

Dr. Warner: No, the initial investment was all covered by research and development funds.

Question: What do you do with the patient who is discharged and then readmitted?

Dr. Warner: We do not have a completely satisfactory way of handling that at the moment. We keep 4,000 patient records in our active file. A patient's file is kept there until his record has completely cleared the record room and all the data have been added to it. We have a system for abstracting information from the written chart in the record room at the time of discharge. The protocols for abstracting this information are set up by the various peer review committees and by the administration. When that information is abstracted and the record is complete, we then put that record on tape. At the present time, we are not readmitting that patient from his tape record at the time he comes in again. We have an average of 85 admissions per day. We just do not go through that tape operation. Instead, when we do have a common hospital number, we can bring all the elements of his file together downstream if that is called for.

Question: Do you have terminals on every ward in the hospital?

Dr. Warner: No, we do not have terminals on every ward. We have 31 ICU beds and these ICU units are heavily instrumented. We have complete records in the computer on these. ICU is all computerized, but the hospital as a whole is not. Every patient has a record in the computer. Every patient except the emergency case, goes through a screening process on admission that includes a self-administered history, an ECG, a pulmonary function, the standard battery of chemistry and hematology tests. That is the routine on admission. Then, all of the subsequent laboratory work goes in. But we do not have a completely computerized medical record.

Question: What computers do you use?

Dr. Warner: We have two Control Data 3300s, with 65,000 (24 bit) words in each.

Question: What kind of business computer do you have and is it dedicated to the hospital job?

Dr. Warner: These two computers communicate with each other. One of them is supported by research; the other is supported by clinical services. The research computer acts as the back-up for the clinical computer. This gives us the reliability level that we need. We also have four minicomputers; some of these are D.E.C., some are Data General, and one is a Control Data 1700. The computer is dedicated to medical applications. We have another IBM computer that is used for all the financial data. These all communicate with each other but the financial data are not kept in our machine. We try to stay clear of that aspect of it.

Question: Is the program machine-dependent?

Dr. Warner: Yes, at this point, it is. We are now putting this into higher level language, as we make the change that I alluded to, of going into a network of minicomputers. All the work on the minicomputer is in the higher level language. I do not know how long it is going to take us to make that conversion. We are not stopping development of new applications in order to make that conversion. It is just sort of evolving as we make the transition in hardware.

119

Reprinted From

COMPUTER APPLICATIONS IN HEALTH CARE DELIVERY

CHARLES WELLER, M.D.
Editor

SYMPOSIA SPECIALISTS / 1460 NE 129 Street / Miami, Fla. 33161 / Phone 891-0118

9

Library of Congress Catalog
Card No. 76-10867
ISBN 0-88372-049-3

Symposia Specialists
MIAMI, FLORIDA, 33161

Printed in the U.S.A.