CHORT : an Original System for Cardiological Database Hospital Reports

D Dee, C Derwael, JL Matton,
R Vanbutsele, C Brohet, M De Kock

Catholic University of Louvain, Medical School, Brussels, Belgium

Abstract

Data collection and management is a tedious and time consuming activity. With CHORT (Cardiac HOspital ReporT), we have designed a new approach to integrating all data related to a specific patient. CHORT allows interactive data input or data acquisition from external systems. CHORT is able to access other local databases.

CHORT can initiate REGAL, a report generator, at any time during a patient's hospital stay. All information is converted into fluent French text before being integrated into the report and merged with free text. REGAL is an interpreted language permitting flexibility in data selection and control over data display and page lay-out.

CHORT improves and speeds up medical file access, suppresses typed report output, and provides a database for clinical and scientific purposes.

1. Introduction

1.1. The problem

A few years ago our Cardiology Department faced quite a problem. At that time, producing a discharge letter took several weeks and this delay was still increasing, due to administrative roundabouts. The idea rose of generating these discharge letters by computer, thereby minimizing manual input and reducing the delay.

1.2. The past

For many years our department had to deal with archiving clinical tests such as vectorcardiograms [1], exercise tests [2], coronary angiograms, etc. These Fortran-applications were all producing computer reports. In 1990, programs and related databases were migrated from a Modcomp Classic system to a HP 9000/835 minicomputer system.

At about the same time, an experiment was started in a PC-environment by implementing a program written in the Clipper language, M-Base (@ trade marks) [3]. This PC-program, running in a LAN-environment, was especially developed for the management of a clinical file system. Although it had several drawbacks, it helped to perform a thorough analysis of the problem.

1.3. The present

Huge amounts of data and past experiences determined the development of a relational database able to deal with complex and heterogeneous medical data. In order to work up the information provided by physicians or by different lab-tests, an easy-to-use computer program was written, allowing fast access to various data.

Also, an original REport GenerAting Language, REGAL, was created to perform immediate selection and conversion of coded patient related data into fluent French.

2. The environment

2.1. Hardware

The core of the computer system is a HP 9000 - G40 minicomputer with 64 MB of RAM and 2 GB of hard disk. The database is backed up daily on a DAT cartridge. About 10 alphanumeric monochrome terminals (HP 700/92) are connected to the computer. In the Cardiology Department, all the networked personal computers running a terminal emulator (AdvanceLink) have also access to the computer by Telnet service. All subunits are provided with at least one terminal or personal computer. The network is Ethernet-based. Four HP Laserjet-III network printers with 2 MB of memory each are connected to the system.

0276-6547/94 $4.00

2.2. Software

The minicomputer is running a Unix operating system (HP-UX 9.04 - 16-user licence). The RDBMS is a HP AllBase f.69 engine. Common Unix tools (awk, nroff, etc.) and standard computer languages (Fortran 77, C-language) are used. The AllBase SQL fulfils the ANSI-II standard. Paper headings are graphic PCL-macro dumps avoiding the use of pre-printed forms. Common Unix profiles specify the user default printer.

Using these standard resources a screen managing program FORM [4,5] was developed, as an interface allowing easy data entry. This interface deals with cursor movement, data format and validity, and also function key management. The number of function keys is limited to 8 in order to keep compatibility with HP-terminals. The interface has also screen dump and time-out facilities. This tool pre-existed, but was further improved while developing CHORT.

An original report generating language was also created, similar to the MERGE-function in word processors, with in addition the possibility to retrieve the appropriate items directly from the database and to convert the coded information into a readable language.

3. Data structure

Each database table corresponds to a logic set of data. Five types of data sets are discerned : administrative, clinical, test and diagnostic data, and code lists. Data are processed per screen or per set of screens.

Administrative data, are entered using the PAT screen. Information about consulting physician and cardiologist is also part of these data.

There are only two screens for clinical data input :

- HOSP : for data related to the hospitalisation (e.g. admission date, discharge date), patient's history, risk factors and current medications;
- CLIN : for data provided by the physical examination.

Test data such as lab, ECG, echo, stress test, coronary angiogram results, etc. can be processed in separate screens. These screens are optional and unlimited in number. In most screens space is available for free text in order to make comments. A free format screen is also provided for results of less common tests.

Diagnostic data complete an hospitalisation :

- EVOL : for complications and comments;
- CONC : deals with the final diagnosis, interventional or medical treatment and follow-up of the patient.

An hospitalisation can only be validated when a conclusion completes the data input, thus HOSP and CONC are mandatory and unique.

One particular table contains the code lists. A code list table has three columns : a unique sequential number (not visible to the user), a mnemonic and the French wording of the mnemonic. Code lists are displayed in browsers. In a screen there are fields ruled by these browsers : only elements from code lists can be put in these fields.

4. The application

4.1. Data and program security

Data can be simply keyed in or entered by automated input. Automated input can inhibit manual update.

Program access and data entry are managed by a system of authorities. This means a double protection. First there is the system security : user ID and password. This gives access to the menu only after detecting CHORT run-authority.

Second, there is an authority management within the program defining four groups of users : administrative, medical, technical and programmer's staff. They have access to particular routines of the program. Authorities are managed at function key level : function keys are only displayed and active, when access is granted. Authority information is kept in separate files and loaded at runtime.

Whenever a set of data is created or updated, a date-, time- and user-ID-stamp is appended and displayed in the right lower corner of the screen.

An intrinsic lock feature is present in the HOSP screen. Entering a reference number in that screen locks all the related sets of medical data. Record numbers of related data sets are stored in an integer array, called an edition vector. This array is generated when the structure of a discharge letter is defined (see paragraph 5.1). Data can still be displayed but not updated.

4.2. Program structure

Essentially there are 4 types of screens. Each of them has a typical set of function keys for navigating through the tree structure of the program. (table 1)

The main screen is the switchboard of the program. It displays the word CHORT and a brief identification of the patient, if one has been selected. The only way to communicate at this level is the use of the function keys. A function key displaying the QUIT option is always present except when consulting browsers.

A second type of screen consists of menus. They are only used twice (choice of input screens and reports). A choice is made by entering a character or by positioning the cursor on the desired item.

Table 1. The 4 screen types with specific function keys

	MAIN	MENU	INPUT	BROWSER
F1	Quit Program	Quit Program	Quit Program	Quit Browser
F2	Medical File		Create Data	Next Screen
F3	Report Generator	List	List	Previous Screen
F4	Patients Administrat.		Update Data	Search
F5	Select Patient	Select Item		Select Item
F6	Physicians List			Top of List
F7			Previous Screen	Bottom of List
F8	Alternate Keyset	Main	Previous Menu	

An input screen, the third type of screen, allows data input and update. On top of each input screen an ID summary of the current patient is displayed. CREATE or UPDATE function keys will only be displayed, if this authority is granted to the user for that particular set of data and if the hospitalisation has not been validated yet.

Manual data entry is quite easy. There are three types of fields in a input screen. *Numeric fields* can have logical limits, and whenever they are exceeded or a wrong format is entered, an error message is displayed. A *list field* can only contain certain pre-defined mnemonics. Such fields can display an exhaustive list, a browser - the last screen type - from which an item can be selected. Only mnemonics from the displayed list can be entered. Finally, *free text fields* allow any alphanumeric input.

Whenever a screen is updated, a validation routine checks possible dependencies between fields, e.g. systolic versus diastolic blood pressure. If there is an inconsistency, an error message is displayed when the CREATE or UPDATE routine is executed. Dependencies are checked between data within a screen, but also consistency with previous entered data is verified.

4.3. Programming logic

A function key stroke activates a particular routine which displays in its turn another screen. For a particular screen two types of library functions control the program. Screen managing functions deal with screen I/O, e.g. field validation and dependencies, and mainly access FORM-routines. Data managing functions access relevant data in the database, performing insert, update, delete, simple and multiple select operations at SQL level. The FORM message routine deals with error handling from both type of functions. *Each data set, say screen, has its particular set of both types of functions.*

5. Reports

When the Report Generator option is activated, one has access to different kinds of reports : several internal forms, interim discharge letter and final discharge letter, the most important one. Output can be a hardcopy or a report display on the screen. At this stage, the default printer and the number of copies can be modified.

5.1. The edition vector

In order to output a discharge letter, an edition vector has to be defined. Report Definition is one of the items of the Report Generator menu. When choosing this menu item, a list of all available data in the period starting at one month before admission and ending at one week after discharge (defined in the HOSP screen), is displayed and can be selected. The report generator takes care of wrapping up the data. *As a result the actual text of the letter does not need to be kept, but an edition vector occupying only a small amount of space in the on-line database is stored instead.* Meanwhile, an ASCII-file is transferred to the hospitals archives.

5.2. REGAL : the report generator

REGAL is an interpreted language. In order to generate a report it needs a preprocessed ASCII-program file and an edition vector. A variable is fully defined by three parameters : the table name, the column name and record number present in the edition vector.

REGAL mainly consists of mere text formatting meta-commands. Interpreter directives allow file inclusion and multi-source code. A conditional construction and a loop instruction enable logical management of the code. Two database meta-commands actually deal with the data to be integrated into the report : the first opens a table for use and the second formats a variable as a simple value or as a complex logical and/or mathematical expression using a C-like syntax.

This allows a flexible report management, thus making possible quite complex French grammatical constructions.

6. Results

The main merits of the program are permanent availability of data and softcopy, report storage requiring only little disk space, keeping report processing at department level, gain in accuracy and legibility, consistency of terminology and uniformity of expressions.

Stereotyped discharge letters are avoided by padding the decoded clinical data with corresponding binding text. The result was a rather pleasant report output in fluent French. Even foreign fellows not mastering the French language are able to output a correct discharge letter.

Patients leave the hospital with an interim discharge letter, yielding the most important elements of their stay in the patient ward. A final report takes about ten days, due to unavailability of test results when patients leave the clinic.

The program has been running for two years. The bonus is a database used for administrative, clinical and research purposes. Today, the database contains 5,500 computerized medical files. Each year 2,000 patients are admitted into the cardiology ward and entered in the database through CHORT. The system deals only with alphanumeric data, implying the use of cheap terminals.

But there is a reverse to every medal. Personnel including interns has to be trained to use the program. The fellows often spend only six months in the patient ward. Moreover, they are not rewarded immediately for their efforts. Data and program maintenance is time consuming and requires the presence of encoders and programmers. The computer needs hardware and software backup, since system access is to be guaranteed 24 hours a day.

7. Conclusion

The purpose of the application was to reduce the delay of discharge letter output. Automated report generation indirectly improved the report delays. Reports are of a good literary quality. There is an important gain in accessibility, accuracy, legibility, consistency and uniformity. One does not need to master the French language completely in order to output a correct discharge letter. Moreover a clinical and scientific database is provided, able to manage information provided by physician or by different labs.

References

[1] Brohet C, Sottiaux B, Derwael-Barchy C, Fesler R, Brasseur L. A computer system for the analysis, display and storage of vectorcardiograms. In: Computers in Cardiology 1977. Long Beach: IEEE Computer Society Press, 1977:585-9.

[2] Detry JM, Vanbutsele R, Derwael-Barchy C, Fesler R, Luwaert R, Robert A, Brohet C. Computer-assisted exercise ECG system based on ST depression, heart rate and Bayesian analysis. Japanese Heart Journal 1982;23:597-9.

[3] De Coster O, Vanbutsele R, Michel C, De Kock M. La place de l'informatique en cardiologie. Partie 3: Informatisation du dossier medical cardiologique. Journal de Cardiologie 1993;5:160-5.

[4] Sansdrap J. X-FORM : a networked HP-1000 form package. In: Proceedings of the International HP-users Conference 1989;TE/NE/03:1-7.

[5] Sansdrap J, Matton JL. T : a package for programming across systems. In: Proceedings of the International HP-users Conference 1991;1003:1-13.

Address for correspondence.

Roland Vanbutsele MS.
Division of Cardiology
Cliniques Universitaires Saint-Luc
Avenue Hippocrate 10/28.81
B-1200 Brussels
Belgium

E-mail address : matton@hedy.ucl.ac.be