To: Distribution

From: Jim Meyerson

Date: 01/17/80

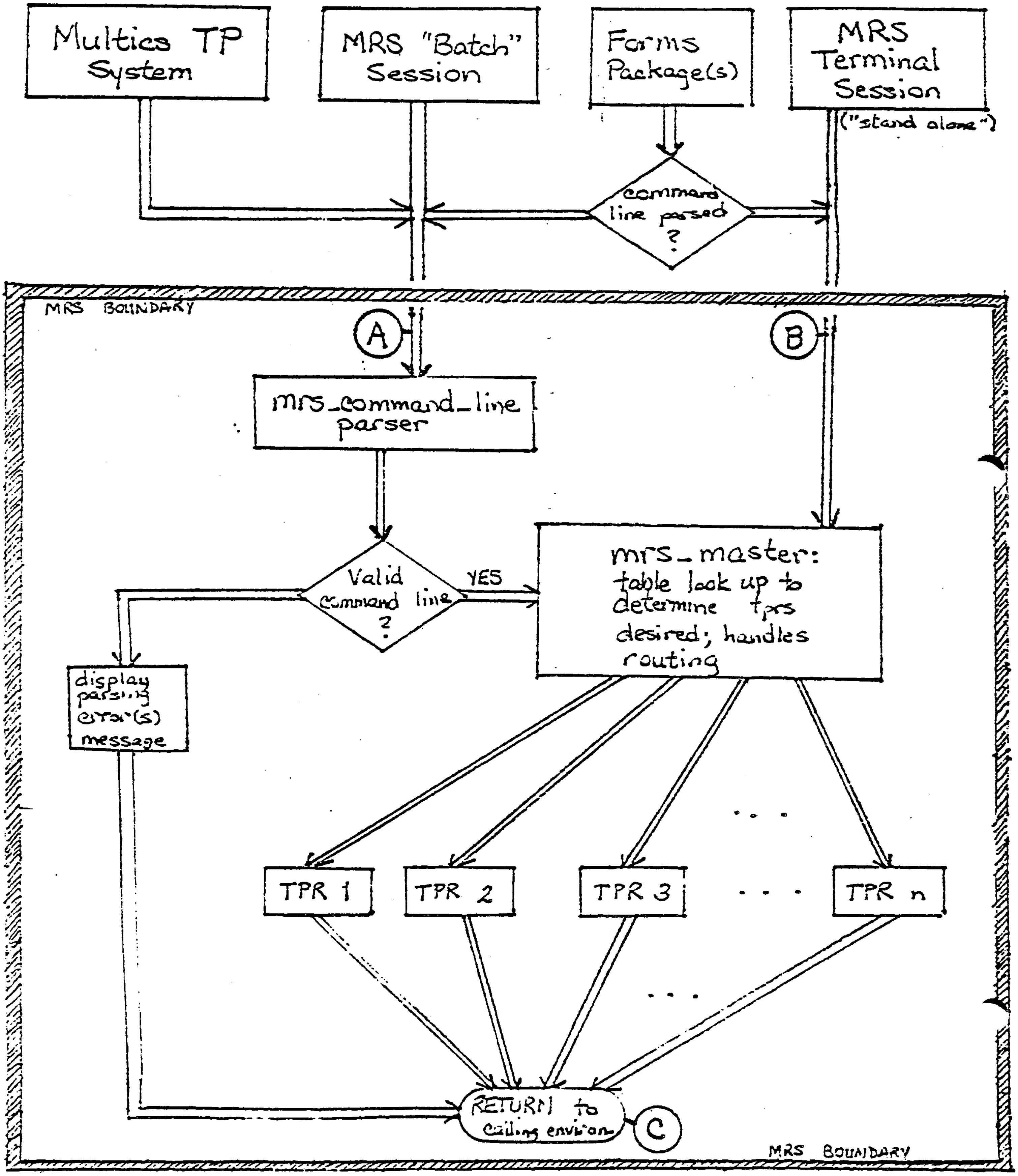
Subject: Medical Records System (MRS)

The Medical Records System(MRS) is an applications package that can be used by doctor offices, health centers, and hospitals to maintain visit schedules, patient records,and staff information. The immediate thrust for the development of an MRS, however, is as a test package for the Multics TP Subsystem, and potentially MRDS and/or vfile_. The MRS maintains an accurate, up-to-date file on all individuals the MRS user has responsibility for. It also protects the confidentiality of the patient information, assuring that unauthorized MRS users do not access personal background information. A computer-stored information base such as MRS, in contrast to data management based on a manual medical records, can be updated from many geographically separate points simultaneously. Information is gathered from precoded forms before entry into the MRS.

DESIGN APPROACH

The design of the MRS comes originally from the design of the Medical Records System currently in use at the Harvard Community Health Plan(HCHP) in Boston, MA. HCHP's MRS is written in standard MUMPS, which is a highly interactive language. Since the current design of the Multics TP Subsystem does not allow interactive processing, the MRS interface has been reworked to allow batching of the requests with the data.

Several design factors were considered most important to the implementation of the Medical Records System. First, MRS was designed to use any number of vfile_ and/or MRDS databases. The database administrator at each individual site may then determine (within certain guidelines) where the data is stored and what


Multics TP System
MRS "Batch" Session
Forms Package(s)
MRS Terminal Session
("stand alone")
command line parsed ?
MRS BOUNDARY
A
B
mrs_command_line parser
Valid command line ?
YES
mrs_master: table look up to determine tprs desired; handles routing
display parsing error(s) message
. . .
TPR 1
TPR 2
TPR 3
. . .
TPR n
. . .
RETURN to calling environ
C
MRS BOUNDARY

Multics project names should have acccess to the information. Second, the system can be easily expanded. To add a new command, two things are required: 1) an entry name is added to the table of available commands, and 2) the TPR to handle the command is added to the MRS.

MRS can operate under the Multics Transaction Processing (TP) subsystem umbrella, as a "background processor" if a forms interface is utilized, or as a stand-alone unit. If the TP Subsystem is being used, then the command names are registered in the tp_command_table, and the MRS is entered by a call from TP to mrs_master. Running stand-alone, MRS "runs itself" through multiple calls of mrs_master. It is understood that for non-TP, non-forms usage, an interactive "menu" interface should exist that will prompt the non-sophisticated MRS user for the appropriate information.

COMMAND LINE FORMATS

An unparsed command line appears as follows:

```
-control_arg1    ARG_VALUE_1   -control_arg2    ARG_VALUE_2    ...
-control_argn ARG_VALUE_n
```

All unparsed input command lines must adhere to the following convention:

1) The number of required control arguments will vary from command to command, but in all cases, at least one control argument and one argument value must be supplied.

2) all control arguments must be immediately preceded by a delimeter (DEFAULT of "-"). Any word that does not begin with a delimeter is considered to be a part of an argument value. (The delimeter is defined at the time the MRS is created.)

3) control arguments must be a single word in length. The PL/1 underscore ("_") is used to concatenate multiple English words into a single word control argument where necessary (i.e. "display_medical_data", "enter_medical_data").

Argument values may contain any number of words as long as the character length of the value does not exceed the maximum line length allowed (DEFAULT of 256 characters). In addition, the entire command line cannot exceed the maximum line length.

Raw input is converted by the MRS parser to a representation that can be handled internally by MRS. If the command line input to MRS is already in the parsed output form, no call is made to the parser. The parser output format is the structure 'arg':

```
dcl 1 arg (no_of_records) aligned, 2 name char(32) varying, 2
value char(256) varying;
```

where no_of_records is a positive fixed decimal(6) value. The arg structure is used to pass user-supplied control arguments from mrs_master to the individual TPRs.

MRS STRUCTURE

The Medical Record System will accept user input in one of two forms. If the command line is in unparsed form as shown by entry A in Figure 1, then MRS will do the command parsing and counting of arguments. If the input to the MRS is already in the form produced by the parsing routine, then entry B is the point where MRS processing begins.

The role of mrs_master is to route the user's command request to the appropriate transaction processing routine (TPR). Within the TPR, the specific tasks desired by the user are performed. For example, if the user desires a display command, the mrs_master routes this command to the correct TPR and within the TPR the database retrieval(s) and data display is done. Once the TPR work is completed, control is returned to the calling environment through a common return sequence (at C).

TERMINOLOGY

In order to maintain confidentiality, a patient's name or social security number is NEVER used for identification by MRS. Instead a unique unit number is created for each patient in the MRS, and a patient's record(s) can be accessed by use of this unit number. When displaying records, again the patient is identified by the unit number.

A unique visit id is created for each visit made by a patient. In most cases, an MRS user does not need to know a visit_id even exists. However, in the event of two visit on the same date by the same patient, the visit_id is the only way to distinguish between visits.

Codes are used in several instances where an English-like value would suffice. This is done for three reasons: 1) it speeds up the inputting of medical data, 2) it creates a more consistent way of describing the diagnosis/procedure, and 3) it saves a substantial amount of storage in the MRS. In all cases, there is a one-to-one correspondence between the code and the longhand ('English') form.

Of particular importance, a medical code is used to identify each diagnosis, therapy, and treatment in the MRS. This is an important control argument for most commands. When displaying records, both the coded and longhand form are shown.

FUNCTIONS AVAILABLE TO THE MRS USER

Commands available within the MRS are defined by combining one of four verbs -- enter, update, display, or display_last -- with the (currently seven) nouns that define database areas (relations). Figure 2 shows these possibilities:

Hence, if a list of all visits by a certain patient is desired, the appropriate command would be found by combining the verb "display" with the noun "visit_data" to get "display_visit_data". An explanation of each of the commands is given in the section COMMANDS.

In addition there are three commands that do not fit the pattern, but are nevertheless needed. These commands are create_visit_data (cvi), list_patient_history(lph), and medical_flowchart(mfc).

The internal naming conventions closely follow the 'algorithm' described above. The enter, display, and display_last TPRs are all named enter_NOUN_, display_NOUN_, and display_last_NOUN_ respectively. Update TPRs are named as an entry into the corresponding enter TPR (enter_NOUN_$update).

WHERE THE INFORMATION IS STORED

As previously mentioned, a combination of vfile_ (Multics virtual file manager) and MRDS (Multics Relational Data Store) databases are used for storing MRS information. vfile_ is quicker, easier to use, and preferred for information that has little dependence on other information in MRS. MRDS has a definate advantage in 'spanning' across information if one set of data is dependent on another set. However, MRDS accesses information substantially slower than vfile_ does.

In the current MRS, personal background information is stored in a vfile_ called 'demographic_data', and is accessed by the unit number key. Information categorized as personal background includes the following information on a patient:

street, city, state, and zip code of residence; date of birth; drug reactions; name and phone number of person to be contacted in an emergency; insurance group number and effective date; name, occupation; serious allergies; primary doctor and/or

MRS COMMANDS

VERB NOUN	enter (e)	display (d)	display_last (dl)	update (u)
background_data (bg)	enter_background_data (ebg)	display_backgound_data (dbg)	—	update_background_data (ubg)
doctor_data (dr)	enter_doctor_data (edr)	display_doctor_data (ddr)	—	update_doctor_data (udr)
health_center (hc)	enter_health_center (ehc)	display_health_center (dhc)	—	update_health_center (uhc)
medical_code (mc)	enter_medical_code (emc)	display_medical_code (dmc)	—	update_medical_code (umc)
medical_data (md)	enter_medical_data (emd)	display_medical_data (dmd)	—	update_medical_data (umd)
surgical_data (sgy)	ester_surgical_data (esgy)	display_surgical_data (dsgy)	display_last_surgery (dlsgy)	update_surgical_data (usgy)
therapy_data (rx)	enter_therapy_data (erx)	display_therapy_data (drx)	display_last_therapy (dlrx)	update_therapy_data (urx)
visit_data (vd)	enter_visit_data (evd)	display_visit_data (dvd)	display_last_visit (dlvd)	update_visit_data (uvd)

Other commands: create_visit_id, enroll, list_patient_history, medical_flowchart nurse; sex; social security number; unit number; work and home phone number.

By using Mutlics access control, access to patient background information is restricted to only a small subset (i.e. doctors) of the MRS user community.

All visit, medical, and internal MRS conversion information is stored in the MRDS database >home_path>mrs_database. The information areas (relations) of the MRDS database are defined as follows:

1. diagnosis information:

a. visit_oriented_medical_data (vomd) is where the diagnosis and treatment information is stored. The attributes (fields) are date*, doctor_code*, med_code*, unit_no*, visit_id* , vomd_status_code, extended_med_code, and comment.

b. supportive services include homemaker, visiting nurse, homecare, etc. The attributes (fields) are unit_no*, date*, and sup_service_code*.

2. administrative information:

a. personal_data is not currently used; this type of information is stored in the
vfile_ 'demographic_data'. However, it is defined here for further expansion. The attributes (fields) of personal_data records are unit_no*, visit_id*, marital_status, sex, race , no_of_children , and occupation.

3. all non-medical information concerning a patient visit:

a. visit_info. Attributes (fields) are date*, doctor_code*, unit_no*, visit_id*, chart_review, dictation, health_ctr_code, hospital_code, time, type_of_visit_code, and comment.

4. staff information: background information on all doctors in MRS is stored here.

a. doctor_info: Attributes (fields) are doctor_code*, name, home_phone, specialty, and work_phone.

b. health_ctr_staff: Attributes (fields) are health_ctr_code* , doctor_code* , and name.

5. conversion (code to English) tables

a. med_code_conversion: Attributes (fields) are med_code* and med_code_longhand.

b. type_of_visit_conversion: Attributes (fields) are type_of_visit_code* and type_of_visit_longhand.

c. hospital_conversion: Attributes (fields) are hospital_code* and hospital_longhand.

d. health_ctr_conversion: Attributes (fields) are health_ctr_code*, health_ctr_longhand, city, main_phone, occupancy, state, street_addr, and zip_code.

e. sup_service_conversion: supportive services conversion. Attributes (fields) are sup_service_code* and sup_service_longhand.

f. vomd_status_conversion: Attributes (fields) are vomd_status_code* and vomd_status_longhand.

* indicates that the attribute (field) is a key for the given relation.

Further information about the specification of the MRDS database can be found by looking at the cmdb (create_mrds_database) file >udd>m>Meyerson>mrs>mrs_database_definition.cmdb .

COMMANDS

The following commands are available to the MRS user:

1. create_visit_id (cvi)

arguments: date and unit number are required; no other arguments accepted.

returns: visit id (that is guaranteed to be uniqe)

creates unique identifier to correspond to each visit entered into the MRS.

2. display_background_data_ (dbg)

arguments: At least one of the following must be supplied: beginning unit number, ending unit number, incremental unit number, or unit number.

returns: personal background information on the patient(s) chosen by the control arguments.

All patients in MRS are identified by a seven digit unit number (no names, social security number, etc., are used for identification). This command returns personal background information on those patient(s) that are specified by the control argument(s).

3. display_doctor_data_ (ddr)

arguments: At least one of the following must be supplied: doctor code(s) and/or doctor name

returns: code, name, home phone, work phone, and specialty for doctor(s) chosen by the control argument(s).

Doctors in the MRS are generally identified by a six character code known (oddly enough) as a doctor code. This command may be used to find the name of the doctor(s) corresponding to certain doctor code(s), or vice versa. In addition, it can be used to find phone numbers where a doctor can be reached.

4. display_health_center (dhc)

arguments: health center code and/or health center name

returns: health center code, name of health center, health center address, main phone for health center, and occupancy of health center.

display_health_center is designed to give relevant information about any health center that is stored in the MRS.

5. display_last_surgery (dlsgy)

arguments: At least one of the following must be supplied: date, after date, before date, doctor code, medical code(s), unit number, or visit id.

returns: date, doctor code(s), medical code(s), unit no(s), visit id(s), medical status, comment(s).

display_last_surgery will display the patient's medical record where the most recent surgery was performed. If date specification is given, then the most recent surgery within the time period is displayed.

6. display_last_therapy (dlrx)

arguments: At least one of the following must be supplied: date, after date, before date, doctor code, medical code(s), unit number, or visit id.

returns: date, doctor code(s), medical code(s), unit no(s), visit id(s),
    medical status, comment(s).

display_last_therapy will display the patient's medical record where the most recent therapy was performed. If date specification is given, then the most recent therapy within the time period is displayed.

7. display_last_visit (dlv)

arguments: At least one of the following must be supplied: date, after date, before date, unit number, or visit id.

returns: visit information.

Based on the control args the user supplies, this command can be used to return the following information:

1) the most recent visit by any patient(optionally during a specified period of time). 2) the most recent visit by a particular patient (optionally during a specified period of time). 3) the last visit on a certain day; the last visit on current day.

8. display_medical_code (dmc)

arguments: medical code(s) and/or English representation of medical code(s).

returns: matching of medical code with English representation.

Medical codes are used to identify each illness, medical procedure, therapy, and medication. This is done for three reasons: 1)it speeds up the inputting of medical data, 2) it creates a more consistent way of describing the diagnosis/procedure, and 3) it saves a substantial amount of storage in the MRS. display_medical_code is used to match the medical code representation with a more meaningful, "longhand" representation.

9. display_medical_data (dmd)

arguments: At least one of the following must be supplied: date, before date, after date, medical code(s), unit number, or visit id.

returns: all medical information requested.

display_medical_data is one of the most heavily used commands. It will return all medical data that satisfies the selection criteria provided by the MRS user.

10. display_surgical_data (dsgy)

arguments: At least one of the following must be supplied: date, before date, after date, medical code(s), unit number, or visit id.

returns: all surgical information requested.

display_surgical_data is very similar to display_medical_data; however, this command will return only surgical information that satisfies the selection criteria provided by the MRS user. (Surgical data is a subset of medical data). In many MRS applications, it is necessary to only provide a subset of information to MRS users in a particular area of work. This gives rise to this command, for it might be desirable to provide the MRS user working in surgery with only surgical information.

11. display_therapy_data (drx)

arguments: At least one of the following must be supplied: date, before date, after date, medical code(s), unit number, or visit id.

returns: all therapy information requested.

display_therapy_data is very similar to display_medical_data; however, this command will return only therapy information that satisfies the selection criteria provided by the MRS user. (Therapy data is a subset of medical data). This command thereby restricts the information MRS users working in therapy can access.

12. display_visit_data (dvd)

arguments: At least one of the following must be supplied: after, before, date, sort, unit number, or visit id.

returns:all visit information satisfied by the selection criteria. If the sort control argument is specified, the records displayed will be sorted based on the sort key (The default is visit id).

13. enroll (enr), enter_background_data (ebg)

arguments: unit number is required. All others are optional: city, comment(s), date of birth, drug reactions, emergency notification, emergency phone, home phone, insurance effective date, insurance group number, name (note this is not required for any identification), primary doctor, primary nurse, serious allergies, sex, social security number (note this is not required for any identification), state, street address, work phone number, zip code.

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

'enroll' is the command by which new patients can be registered into the MRS. If personal background information on the patient(s) is already stored in the MRS, then an error message will be printed, and no entering/updating of the affected record(s) will be done.

14. enter_doctor_data (edr)

arguments: doctor code(s) and/or doctor name required. All others optional: doctor's home phone number, doctor's work phone number, doctor's specialty.

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

enter_doctor_data is used to enter information about the doctors on the staff. Relevant information includes doctor name (the doctors are identified by doctor code within the MRS), phone numbers where the doctor can be reached, and a listing of the doctor's medical specialty. If information is already stored in MRS on the doctor chosen by the control argument(s), then an error message is displayed, and no entering or updating of records is done.

15. enter_health_center (ehc)

arguments: health center code is required. All others optional: English name for health center, street address, city, state, zip code, main phone, occupancy.

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

This command allows the entering of address, phone number, and occupancy information on any of the health centers, hospitals, and doctor offices that use the MRS. If information already is stored on the health center, then no entering or updating of information is done, and an error message is displayed.

16. enter_medical_code (emc)

arguments: medical code, English description of medical code (both are required).

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

Medical codes are used to form a standardized way of explaining medication, medical procedures, and diagnoses. They are found on the Data Entry Forms (see Appendix), so in general, there is no need to look up a specific code. In display routines, however, both the medical code and its longhand description are given. For this reason, it may be desirable to enter the appropriate 'conversions' with this command. If the medical code is already stored in the MRS, the record(s) associated with the medical code are updated.

17. enter_medical_data (emd)

arguments: date, doctor code, medical code(s), unit number, and visit id are required. The rest are optional: medical status, extended medical code(s), comment(s).

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

enter_medical_data is another heavily used command. It is used whenever new medical information on a patient is to be stored in the MRS. If information is already stored in the MRS with the exact same selection criteria (combination of date, medical code, unit number, and visit id), then an error message is printed and no updating or entering of new records is done. If the selection criteria matches, but no visit id is given, then a new, unique visit id is generated.

18. enter_surgical_data (esgy)

arguments: date, doctor code, medical code(s), unit number, and visit id are required. The rest are optional: medical status, extended medical code(s), comment(s).

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

enter_surgical_data is used in much the same manner as 'enter_medical_data' to enter information about patient(s)' surgery. If a MRS user has enough access, enter_medical_data can be used instead of enter_surgical_data. However, this command must be used if the MRS user (who presumably works in surgery) does not have sufficient access to other information. If information is already stored in the MRS with the exact same selection criteria (combination of date, medical code, unit number, and visit id), then an error message is printed and no updating or entering of new records is done. If the selection criteria matches, but no visit id is given, then a new, unique visit id is generated.

19. enter_therapy_data (erx)

arguments: date, doctor code, medical code(s), unit number, and visit id are required. The rest are optional: medical status, extended medical code(s), comment(s).

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

enter_therapy_data is used in much the same manner as 'enter_medical_data' to enter information about patient(s)' therapy. If a MRS user has enough access, enter_medical_data can be used instead of enter_therapy_data. However, this command must be used if the MRS user (who presumably works in therapy) does not have sufficient access to other information. If information is already stored in the MRS with the exact same selection criteria (combination of date, medical code, unit number, and visit id), then an error message is printed and no updating or entering of new records is done. If the selection criteria matches, but no visit id is given, then a new, unique visit id is generated.

20. enter_visit_data (evd)

arguments: date, unit number, visit id are required. All others are optional: indicator if chart review made of visit, indicator if dictation was made following visit, health center or hospital where visit made, time of visit, type of visit, comment(s).

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

enter_visit_data is a command that can be used in one of two ways. First, it can be used as a daily calendar, to schedule upcoming patient visits. Also, it can be used to enter non-medical information gathered on a patient immediately following his/her visit.

21. list_patient_history (lph)

arguments: unit number of patient is required. All others are optional: date, doctor code, medical code(s), and visit id. In addition, the MRS user can specify how the records are to be sorted before they are displayed.

returns: all medical data that satisfies the selection criteria provided by the MRS user.

list_patient_history can be used to give a complete medical history on a patient. The default is for all medical information to be displayed, however, by using date, medical code(s), and/or visit id parameters, the amount of information displayed can be limited. Unless a sort criteria is supplied by the user, records are listed with the most recent record (chronologically) displayed first. If sort criteria is supplied, then the records will be sorted according to the sort key.

This command is identical to the 'medical_flowchart' command, except that 'medical_flowchart' does not require a unit number, and can be used to trace the occurences of medical codes among many patients, or the occurences during a range of dates.

22. medical_flowchart (mfc)

arguments: One of the following control arguments is required: date, doctor code, medical code(s), unit number, and visit id. In addition, the MRS user can specify how the records are to be sorted before they are displayed.

returns: all medical data that satisfies the selection criteria provided by the MRS user.

medical_flowchart can be used to give 1) a complete medical history on a patient; 2) a list of all visits on a date or set of dates; 3) a list of all patients with a set of medical codes; and many other types of information. The default is for all medical information to be displayed, however, by using date, medical code(s), and/or visit id parameters, the amount of information displayed can be limited. Unless a sort criteria is supplied by the user, records are listed with the most recent record (chronologically) displayed first. If sort criteria is supplied, then the records will be sorted according to the sort key.

23. update_background_data (ubg)

arguments: unit number is required. All others are optional: city, comment(s), date of birth, drug reactions, emergency notification, emergency phone, home phone, insurance effective date, insurance group number, name (note this is not required for any identification), primary doctor, primary nurse, serious allergies, sex, social security number (note this is not required for any identification), state, street address, work phone number, zip code.

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

update_background_data is used to change personal background information on patient(s) in the MRS. If information has not been previously stored on a patient, then a record will be entered on the patient. If a record is found in MRS, then it will be updated with the new information provided.

24. update_doctor_data (udr)

arguments: doctor code(s) and/or doctor name required. All others optional: doctor's home phone number, doctor's work phone number, doctor's specialty.

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

update_doctor_data is used to enter or update information about a doctor on the staff. Relevant information includes doctor name (Each doctor is identified by doctor code within the MRS), phone numbers where the doctor can be reached, and a listing of the doctor's medical specialty. If information is already stored in MRS on the doctor chosen by the control argument(s), then the affected record is modified to include the new information. If no information on the doctor is found in the MRS, then a new record is created.

25. update_health_center (uhc)

arguments: health center code is required.. All others optional: English name for health center, street address, city, state, zip code, main phone, occupancy.

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

This command allows the entering or updating of address, phone number, and occupancy information about any of the health centers, hospitals, and doctor offices that use the MRS. If information on the health center is found in the MRS, then the appropriate record(s) is/are updated. If no information exists in the MRS, then new records are added.

26. update_medical_code (umc)

arguments: medical code, English description of medical code (Both are required)

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

Medical codes are used to form a standardized way of explaining medication, medical procedures, and diagnoses. They are found on the Data Entry Forms (see Appendix), so in general, there is no need to look up a specific code. In display routines, however, both the medical code and its longhand description are given. For this reason, it may be desirable to enter the appropriate 'conversions' with this command. If the medical code is already stored in the MRS, the record(s) associated with the medical code are updated. If the medical code is not stored in MRS, then a new record is created and stored.

27. update_medical_data (umd)

arguments: date, doctor code, medical code(s), unit number, and visit id are required. The rest are optional: medical status, extended medical code(s), comment(s).

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

update_medical_data is another heavily used command. It is used whenever new or updated medical information on a patient is to be stored in the MRS. If information is already stored in the MRS with the exact same selection criteria (combination of date, medical code, unit number, and visit id), then the record(s) is/are updated to reflect the new information. If the selection criteria matches, but no visit id is given, then a new, unique visit id is generated. If no information is found in the MRS that satisfies the criteria, then a new record(s) are added to the MRS.

28. update_surgical_data (usd)

arguments: date, doctor code, medical code(s), unit number, and visit id are required. The rest are optional: medical status, extended medical code(s), comment(s).

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered/updated is displayed.

update_surgical_data is used in much the same manner as 'update_medical_data' to enter or update information about a patient(s)' surgery. If an MRS user has sufficient access, update_medical_data can be used instead of update_surgical_data. However, this command must be used if the MRS user (who presumably works in surgery) does not have sufficient access to other information. If information is already stored in the MRS with the exact same selection criteria (combination of date, medical code, unit number, and visit id), then the record(s) affected is/are updated to include the new information. If the selection criteria matches, but no visit id is given, then a new, unique visit id is generated. If no information is found for the criteria given, then a new record is created.

29. update_therapy_data (urx)

arguments: date, doctor code, medical code(s), unit number, and visit id are required. The rest are optional: medical status, extended medical code(s), comment(s).

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered/updated is displayed.

update_therapy_data is used in much the same manner as 'update_medical_data' to enter or update information about patient(s)' therapy. If an MRS user has sufficient access, update_medical_data can be used instead of update_therapy_data. However, this command must be used if the MRS user(who presumably works in therapy) does not have sufficient access to other information.If information is already stored in the MRS with the exact same selection criteria (combination of date, medical code, unit number, and visit id), then the record(s) affected is/are updated to include the new information. If the selection criteria matches, but no visit id is given, then a new, unique visit id is generated. If no information is found for the criteria given, then a new record is created.

30. update_visit_data (uvd)

arguments: date, unit number, visit id are required. All others are optional: indicator if chart review made of visit, indicator if dictation was made following visit, health center or hospital where visit made, time of visit, type of visit, comment(s).

returns: The information returned depends on the start-up parameter 'display_record'. If display_record is off ("0"b), then no results are displayed (except error messages). If display_record is on ("1"b), then a copy of the record entered is displayed.

update_visit_data is a command that can be used in one of two ways. First, it can be used as a daily calendar, to schedule upcoming patient visits. Also, it can be used to enter or update non-medical information gathered on a patient following his/her visit.

MRS START-UP

In order to create a MRS environment, the following must be done (in the sequence shown):

1. Define MRDS cmdb segment(s) describing the MRDS database structure(s) for storage of MRS information. An example of the current default MRS database set-up can be found in the segment >udd>m>Meyerson>mrs>mrs_database_definition.cmdb .

2. Define switch name(s) to vfile_ database(s) to be used by the MRS. This is done by assigning vfile_swN a value in dcl_mrs_start_up_params.incl.pl1 . Default is [wd]>demographic_data.

3. Assign values to user-settable start_up parameters. Parameters that can be changed are all found in the include file >udd>m>Meyerson>mrs>dcl_mrs_start_up_params.incl.pl1 . They include MRDS database paths, vfile_ switch name(s), maximum number of MRS records to be retrieved at one time(DEFAULT = 50), default phone area code(DEFAULT = "(617)"), command line delimeter(DEFAULT ="-"), and the storage mode("upper", "lower", or "mixed" characters, DEFAULT = "upper"). In addition, flags can be set to determine whether warning and error diagnostics are to be displayed if a TPR does not complete successfully.

4. If any of the DEFAULT start-up parameters have been changed, recompile the MRS to include the new values.

5. Using either the linus store command or mrds_call stores, load the MRS databases with conversion values (affecting type_of_visit_conversion, med_code_conversion, health_ctr_conversion, doctor_code_conversion, doctor_info_, vomd_status_conversion relations).

6. If using TP, enter TPR entry names into tp_command_table and create the linkage between user commands and MRS TPRs. If using forms package as stand-alone, write/install background software to take screen data and convert it into arg structure or command_line format (if necessary). If using MRS as stand-alone, entry names are already 'registered' in mrs_master.

7. If using TP, go through TP Start-Up procedure. If using MRS as a stand-alone, type MRS to begin execution. There is no special procedure associated with the storage, retrieval, etc., of the first record. If using Forms, go through any necessary Forms start_up procedures.

8. Pat yourself on the back and proceed with caution. You have now created a Medical Records System.

ADDING MORE COMMANDS

The Medical Records System is designed so that any number of new commands can be added to it, depending on the site's specific needs. TPRs may be written in PL/1, COBOL, or any other language that supports the Multics calling convention. Once the TPR has been written and debugged, it is entered into the MRS by adding its name and entry point to the appropriate program or table. When using TP, the entry point and command name(s) are entered into the tp_command_table. If using MRS as a stand-alone then the appropriate routing seuqence should be written into mrs_call_command in the form of

```
     if command_name = "new_TPR" then call new_TPR(dbi_mrs,
total_arg_count, arg, code);
```

This is all that is required.

CONCLUSION

This is an overview of the Medical Records System. The system as currently implemented takes advantage of the relational properties of MRDS, to make 'cross-referencing' of medical information. It does not contain billing functions, but should otherwise be able to replace the filing, posting, and scheduling functions currently handled manually in most health centers and doctor offices. Paper records suffer from a severe limitation in that the physical record can be in only one place at a time. In contrast, if the information is stored online, it can be displayed and/or updated in many places simultaneously. The problems and costs of filing, transporting, and updating are thus removed.

MRS also offers the opportunity to compile statistics on the information stored in a much quicker fashion, than sifting through paper records. Its interface is (hopefully) simple and easy to pick up so that non-DP personnel will not experience the usual fears of computers.

GLOSSARY

The following is a list of all the control arguments available to the MRS user:

-after DT, -aft DT

used by display and display_last commands. Selects all applicable medical records later in date (i.e. closer to present) than date given as argument.

-before DT, -bef DT

used by display and display_last commands. Selects all applicable medical records earlier in date (i.e. farther from present) than date given as argument.

-beginning_unit_number N, -beginning_unit_no N, -bun N

used by display and display_last commands. Selects all applicable medical records with unit numbers larger than the unit number given as argument.

-chart_review CHAR, -chart CHAR

used by commands involving personal background data, chart_review is a single character -- 'y'(yes) or 'n'(no) -- to indicate if a chart review by the patient's doctor was made following a visit.

-city STR

used by commands involving personal background data -- name of city, town, village, etc.

-comment STR, -comm STR

used by most commands; field for remarks, or information that is not provided in other fields.

-date DT, -dt DT

used by commands involving visit or medical information. Example: January 20, 1980 (MRS Key)

-dictation CHAR, -dict CHAR

used by commands involving personal background information; single character -- 'y' (yes) or 'n' (no) -- to indicate if doctor dictated additional information not entered into MRS following patient visit.

-date_of_birth DT -dob DT

used by commands involving personal background information. Control argument gives patient's date of birth. Form is same as -date.

-doctor_code STR, -dr_code STR, -drc STR

used by commands that involve visit, medical, or doctor information. MRS-defined shorthand representation of each MD in the MRS.

-doctor_longhand STR, -dr_longhand STR, -drlh STR

used by commands that involve doctor information. This is the English representation for each doctor code.

-drug_reactions STR, -reactions STR

used by commands involving personal background information; field to explain any reaction(s) patient may have to medication or treatment.

-emergency_notification STR, -emerg_notif STR, -en STR

used by commands involving personal background information; full name of person(i.e. closest relative) to be contacted in case of emergency to patient.

-emergency_phone STR, -emerg_phone STR, -ep STR

used by commands involving personal background information; phone number (including area code) of person specified by -emergency_notification.

-extended_med_code STR, -emc STR

used by all commands that involve medical information; field to be used if -medical_code is not specific enough to fully identify diagnosis or treatment.

-from_date DT, -from DT

used by display and display_last commands. Selects all applicable medical records later in date (i.e. closer to present) that date given as argument.

-health_center_code STR, -health_ctr_code STR, -hcc STR used by commands involving visit or health center information; MRS-defined shorthand for each doctor's office, community health center, and hospital for which MRS information is stored.

-health_center_longhand STR, -health_ctr_longhand STR, -hc_longhand STR, \cSTR

used by commands involving visit or health center information; English name representing each doctor office, community health center, and hospital in the MRS (corresponding to each health_ctr_code).

-home_phone STR

used by commands involving personal background or doctor information. The argument gives home phone number.

-hospital_code STR, -hosp_code STR

used by commands involving hospital, hospital_code is a MRS-defined shorthand for each hospital in MRS.

-hospital_longhand STR, -hosp_longhand STR, -hosplh STR

used by commmands involving hospital information, this control argument gives English name for each hospital in MRS (There is a one-to-one correspondence between each hospital_code and hospital_longhand).

-insurance_effective_date DT -ins_eff_date DT, -ied DT

used by commands involving personal background information; date of issuance of insurance policy(s) covering patient.

-insurance_group_number STR, -ins_group_no STR, -ign STR

used by commands involving personal background information; number of company policy or health group number, if applicable.

-main_phone STR

switchboard phone number for office or hospital.

-marital_status STR

not currently used; single, married, divorced, widowed, separated, etc. marital status of patient

-medical_code STR, -med_code STR, -mc STR used by all commands involving medical information; MRS-defined shorthand to uniquely identify each diagnosis, treatment, or therapy to any patient in the MRS. (MRS Key).

-medical_code_longhand STR, -med_code_longhand STR, -mc_longhand STR, -mclh STR

used by all commands involving medical information; English description of each medical_code stored in the MRS.

-name STR

patient name. This field is not used for identification anywhere in the MRS, and the information is only available to priviledged users who have access to the personal background information.

-number_of_children N, -no_of_children N

number of children patient has. Not currently used.

-occupancy STR

used by commands involving health center, doctor, or hospital information; number of people a particular health center or hospital is set up to handle.

-occupation STR, -occ STR

used by commands involving personal background information; type of work patient is employed in.

-primary_doctor STR, -primary_dr STR, -pdr STR

used by commands involving personal background information; name of medical doctor normally seen by patient first.

-primary_nurse STR, -primary_rn STR, -prn STR

used by commands involving personal background information; name of nurse patient normally sees first (applicable with community health centers).

-serious_allergies STR, -allergies STR, -algy STR

used by commands involving personal background information; field to list any patient reactions to pollens, dusts, foods, etc.

-sex CHAR

used by commands involving personal background information; single character -- 'm' (male) or 'f' (female) -- to indicate sex of patient.

-specialty STR, -spec STR

doctor's area of expertise. This control argument is used by all commands involving doctor information.

-social_security_number STR, -ssn STR

patient's social security number. This value is not used for identification anywhere in MRS, and is available to only privileged MRS users (who have full acccess to personal background data).

-state STR, -st STR

used in most non-display commands; abbreviation for state, province, or territory patient resides in.

-street_address STR, -street_addr STR, -addr STR

used by commands involving personal background information; postal address (p.o. boxes may be used).

-status_code STR, -vomd_status_code STR, -sc STR

used by all commands involving medical information; shorthand form to explain 'seriousness' of medical illness, surgery, etc., status_code should be used only when a medical_code has been specified.

-status_longhand STR -vomd_status_longhand STR, -slh STR

used by all commands involving medical information; English representation of status_code.

-supportive_service_code STR, -sup_service_code STR

MRS-defined shorthand for describing any non-medical aids given patient (such as crutches, home nurse, etc.). This is a seldom-used argument utilized by those commands that involve supportive services.

-supportive_service_longhand STR, -sup_service_longhand STR

English representation for each supportive_service_code. This is a seldom-used argument utilized by those commands that involve supportive services.

-time TM, -tm TM

time of day. All times within MRS are expressed in military time ("24-hour clock"), however, the user may supply times with 'am' or 'pm' appended in addition to military time. If am or pm is not specified, am is assumed. If no time zone is given, the computer system's default time zone (i.e. mst for System M) is assumed. This argument is used by commands that involve visit information.

-type_of_visit_code STR, -tov STR

MRS-defined shorthand to explain the way in which patient visit was arranged. Possible values include 'c' (call-in), 'e' (emergency), 's'(scheduled), 'w' (walk-in), etc. This argument is used by commands that involve visit information.

-type_of_visit_longhand STR, -tov_longhand STR, -tovlh STR

English representation for each type_of_visit_code. This argument is used by commands that involve visit information.

-unit_number N, -unit_no N, -un N

most commonly used control argument; used by all commands that involve any information about a patient; unique identifier for each patient entered into the MRS. (MRS Key)

-visit_id N, -vid N

used by all commands that involve visit or medical information, a visit_id is the unique identifier for each visit record stored in the MRS. (MRS Key)

-work_phone STR

used by commands involving personal background information; phone number (including area code) where patient or doctor can be reached at his place of employment.

-zip_code N, -zc N

postal zip code. Used by many commands.