Abstract

This project is the creation of a software interface that provides a link between Unix word-processing programs and the Medical Manager® software package. This interface provides additional options for accessing, manipulating, and storing word-processing files while the user is logged into the Medical Manager®. Options include balancing and remapping the word-processing pseudo drives, creating and accessing word-processing document templates, and tracking multiple word-processing files from individual patient accounts.
Table of Contents

ABSTRACT ................................................................................................................ ii
TABLE OF CONTENTS ............................................................................................ iii
LIST OF FIGURES .................................................................................................... iv
LIST OF TABLES ...................................................................................................... v
BACKGROUND AND RATIONALE ........................................................................... 1
NARRATIVE .............................................................................................................. 4
ENVIRONMENT ......................................................................................................... 9
PROCEDURE ........................................................................................................... 10
END RESULTS ......................................................................................................... 27
REFERENCES .......................................................................................................... 28
APPENDIX ............................................................................................................... 29
List of Figures

FIGURE 1-INTERFACE FLOW CHART..........................................................11
FIGURE 2-FILE LOCATIONS.................................................................12
FIGURE 3-WORD-PROCESSING ID.......................................................13
FIGURE 4-DML FLOW CHART..............................................................19
FIGURE 5-SCRIPTFILE.....................................................................22
FIGURE 6-SOFTWARE DIALOGUE.......................................................25
List of Tables

TABLE 1-PATIENT FILE.................................................................15
TABLE 2-CONTROL FILE..............................................................17
TABLE 3-PATIENT RECORDS..........................................................23
Background and Rationale

Since its development in 1982, more physicians have relied on the Medical Manager® software than any other system to meet the demands of an ever-changing health-care market. Today health-care providers nationwide recognize The Medical Manager® software as the leading system for physician office and physician network automation. Installed in over 24,000 client sites and used by more than 110,000 physicians, The Medical Manager® software is the most trusted and most proven system available. The Medical Manager® software's base package serves as the foundation of the system and includes many of the functions needed for a complete practice-management system.

The Medical Manager® software, which is used by doctor's offices and hospitals to track patient account information, billing, and other office related functions does not include a word-processing module. Since there are other word-processing software packages which run on an Unix operating system, it is beneficial to have a way to incorporate the word-processing programs and the Medical Manager® efficiently. This project is designed to create an interface between word-processing programs and the Medical Manager®.

Currently, the word-processing files are tracked by the Medical Manager® software in predefined Unix file directories. As new patient accounts are added to the system, all associated word-processing files are placed in the same directory. When one directory becomes excessively full of files, the system processing speed slows down drastically.
Periodically, files must be manually redistributed to different pseudo drives to try to increase system efficiency. Once the word-processing files are redistributed, the Medical Manager® data files must also be updated with the new word-processing directory information for each account. Since this is an ongoing problem, because new patient accounts are constantly added to the system, it is very beneficial to have an automated way to handle this problem.

Also, when a letter or document is created using a word-processing program for one patient, there is a very high probability that the same or a similar document will also be needed for several other patient accounts. Rather than having to recreate the entire document for each new patient account, it would be much more efficient to have the ability to save the original letter with a descriptive title. The user would then have the ability to choose from existing document templates and add each one as a separate word-processing file for each patient account. By only changing a few items such as patient name and demographics, new documents will be created very easily.

Currently, the Medical Manager® allows for only one word-processing file to be associated with each patient account, and this is not enough. There can often be several letters, transcription notes, and other important documents needed per patient account. By incorporating a way to track multiple word-processing files per patient account and link them back to the Medical Manager® patient account file, the user is able to have access to each document by entering only the patient account number. This alleviates the limitation of being able to associate only one word-processing file per patient account.
With such a powerful office-management software system, the addition of the ability to interface word-processing programs from within the system greatly enhances the functionality of the Medical Manager® system. It also helps current users of the Medical Manager® track and utilize their existing word-processing programs more easily.
Narrative

This project is the creation of a software interface which provides a link between Unix word-processing programs and the Medical Manager® software package. The interface provides options for accessing, manipulating, and storing word-processing files while the user is logged into the Medical Manager®. Options include remapping the word-processing pseudo drives, creating word-processing documents from templates, and tracking multiple word-processing files from within individual patient accounts in Medical Manager®.

The option to remap the word-processing pseudo drives takes all current word-processing files associated with Medical Manager® patient accounts and rearranges the files in the operating system file directory structure to ease the system load. Medical Manager® contains word-processing drive information in a special system control file. Through the interface, the user enters the specifics about the available word-processing drive locations currently on the system. An Unix operating system script using Perl counts all word-processing files associated with the Medical Manager® accounts and divides the files by the number of word-processing drives minus one. The final word-processing drive does not contain any files because it is used for all new patient accounts. Any new directories are created by the Medical Manager® software when they are added to the list of pseudo drives. This accommodates any new word-processing drives added to the system. The word-processing files are then redistributed to the remaining word-processing pseudo drives. Once all files have been relocated, the Medical Manager® patient file is updated with the new word-processing drive letter for each account. The Medical Manager control
file is also updated to include the additional word-processing drives, each new Unix file directory, and the patient-account-number range for each new directory.

The user accesses a shell script menu which allows the user to choose any of the word processing interface options. This logs the user into the Medical Manager® and executes a custom menu option using an automated jobstream program to automatically access the Medical Manager® and the correct custom menu option. The first choice on the shell script menu allows the user to begin the remapping of the word-processing pseudo drives. Once this is selected, the system then verifies that all users but one are logged out of the entire system. If all other users are not out, then a message appears listing all users in the system and warning the user that everyone must be logged out and execution will stop. The user is returned to the shell script main menu. If all users are out of the system, then the script gives control to the Medical Manager® custom menu option which prompts the user for a password specific to this custom menu option only. Executing this menu option requires a special password. Once the correct password is entered, the user is prompted for the current word-processing pseudo drive information. The user enters the current range of word-processing pseudo drives. The system begins remapping all word-processing files, and once completed, the user receives a display option containing the new mapping of the word-processing files and the current word-processing drive that is to be used for all new accounts. The user is then given an option to print the information to the system default printer which has been updated in the Medical Manager® control file. The user is then exited back to the shell script main menu and given a chance to choose another option or exit the script completely and log back in to the Medical
Another option of the word-processing interface is an option to catalogue the word-processing document templates. The user has the option through a Unix shell script to name a current word-processing document with a descriptive title and save it. This file can be accessed when needed for use with other patient accounts. The user can manually fill in all patient demographic information or use the appropriate script to import fields from the patient information records from Medical Manager® data files into the word-processing document. Once created, the file is saved with a title which can be accessed for use with other patient accounts, and the system copies the file to the correct word-processing pseudo drive for the current patient account.

If the option to save a word-processing document template is chosen, then the user is prompted to give a current word-processing file template number or search a list of files in a specific directory. The script verifies that the file exists. If the file chosen does not exist, the user is given an error message on the screen and taken back to the shell script main menu. If the document does exist, the user is prompted to enter a descriptive label for the document, and the file is saved under this name. All of the word-processing template documents are loaded in a special template directory. Once the document template is created, the user is given the option to enter an account number to use with the document template. If no account number is entered, then the user is taken back to the shell script main menu and asked to choose another option or exit back to Medical Manager®.
This option of the interface also gives the user the option of merging the patient account information with the document template. The user selects the option and enters the patient account number and chooses one of the patient's visits. The DML program then passes all the patient information into a temporary file for the word-processing program to use to merge into the document. The script for this option creates a subdirectory for each account by the date of service of the visit selected, and if it does not already exist, the directory is created.

The final option on the shell script menu gives the user the option to track multiple word-processing files per patient account. The script prompts the user for an account number which contains multiple word-processing files. Once the account number is entered, a separate subdirectory is created for this account and the Medical Manager® patient file updated with the new word-processing path for this patient. The user is then prompted for another patient account or the option to return to the shell script main menu.

The entire interface is controlled by a Korn shell script that is executed from outside of the Medical Manager® but with direct access to the Medical Manager® custom menu screens through the use of a jobstream program. The script provides an easy-to-use menu system that invokes additional prompts for each option selected by the user from the menu. The DML (data merge language) is used to update or access all Medical Manager® data files, while the program written in Perl runs all operating-system file manipulations. A password on the custom menu options is used to control access to
certain files that are being used concurrently by other users of Medical Manager®.

Once the user has completed all desired options, then the option to exit the word-processing interface is selected. The user is taken back to the Medical Manager® custom menu screen and the script is exited.
Environment

The operating system that is used is Unix Sco Open Server 5. The Medical Manager® software and the word-processing program are both running under this platform. The operating system uses the Korn shell. DML (data merge language) and Perl are the programming languages that are used. The hardware capabilities must be an Intel Pentium server with at least 233 MHz and 32 megabytes of memory. There should also be at least 2 gigabytes of hard drive space available for all software components and data files.
Procedure

This interface provides options for accessing, manipulating, and storing word-processing files while the user is logged into the Medical Manager®. The options of the interface are remapping the word-processing pseudo drives, creating word-processing documents from templates, and tracking multiple word-processing files from within individual patient accounts in Medical Manager®.

The procedure for this project was developed using a top-down design [4]. Each separate module leads to its own set of programs that can run as separate programs. Figure 1 details the design for the interface between the Medical Manager® and the word-processing program. The interface provides three options, each of which perform a different function. All options, once complete, return to the main script for either exit from the script or the user may select another option and continue.

The entire interface was designed to provide quick access to the options for incorporation of the Medical Manager® database and the Unix word-processing program. Each option executes the specified programs once requested, and returns control to the shell script that controls the interface. Each module or option of the interface can be run independently and can function as a stand-alone program. There are four separate options in the interface. As shown in Figure 1, option 1 executes the remap of the word-processing drives. Option 2 executes the word-processing document template, and option 3 allows
for multiple word-processing files per each patient account. Option 4, the final option exits the interface and returns to the Medical Manager® software.
The option to remap the word-processing pseudo drives is executed from a shell script which starts a Medical Manager® jobstream program that logs the user into the Medical Manager® and changes menus to a custom menu command line. The custom menu command line executes a DML program that begins the remap of the files. This option first prompts the user for the range of pseudo drives that are to be used for the remapping of the files. The DML program creates a temporary text file containing all of the pseudo drives and their corresponding Unix path. These files are separated in a subdirectory structure defined in Figure 2, which shows the control.dat file locations [1].

Using File Locations:

At the Install Options Menu, select option - 4 - File Locations. A screen similar to the following appears:

<table>
<thead>
<tr>
<th>Location</th>
<th>Corresponding Directory Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>(current working directory)</td>
</tr>
<tr>
<td>A</td>
<td>/usr2/meddata/</td>
</tr>
<tr>
<td>B</td>
<td>/usr2/medprogs/</td>
</tr>
<tr>
<td>C</td>
<td>/usr2/medrpg/</td>
</tr>
<tr>
<td>D</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>

Enter (N)ext, (P)revious, or (M)odify : .

Figure 2 – File Locations