THE FRAME-BASED MODEL OF NURSING CARE PLANS
ON THE HELP SYSTEM

By

Ping Wang

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Medical Informatics
The University of Utah
December 1989
THE UNIVERSITY OF UTAH GRADUATE SCHOOL

SUPERVISORY COMMITTEE APPROVAL

of a thesis submitted by

Ping Wang

This thesis has been read by each member of the following supervisory committee and by majority vote has been found to be satisfactory.

[Signatures]

Chair:
To the Graduate Council of the University of Utah:

I have read the thesis of Ping Wang in its final form and have found that (1) its format, citations and bibliographic style are consistent and acceptable; (2) its illustrative materials including figures, tables and charts are in place; and (3) the final manuscript is satisfactory to the supervisory committee and is ready for submission to The Graduate School.

Chair, Supervisory Committee

Approved for the Major Department

Homer R. Warner
Chair/Dean

Approved for the Graduate Council
ABSTRACT

The computerized nursing care plan is an integral part of HELP nursing subsystem. The frame technology, which is a new architecture for implementing a knowledge driven hospital information system, has been used to computerize the nursing care plan on the HELP system. Different methods of data retrieval and storage have been introduced, tested and compared in terms of programming speed and structure. A general and flexible model called "the frame-based model" has been developed. The model is flexible enough to accommodate the change of the nursing care plan database and it could be a general model for other applications on the HELP system. The model is able to integrate HELP decision support functions so that the expert knowledge can be used in the automation of the nursing care plan.
To My Parents
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>x</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>xi</td>
</tr>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>The Nursing Care Plan</td>
<td>1</td>
</tr>
<tr>
<td>Computerization</td>
<td>4</td>
</tr>
<tr>
<td>The HELP System</td>
<td>5</td>
</tr>
<tr>
<td>The HELP Nursing Subsystem</td>
<td>9</td>
</tr>
<tr>
<td>The Research Goals</td>
<td>11</td>
</tr>
<tr>
<td>II. COMPUTERIZED NURSING CARE PLANS</td>
<td>13</td>
</tr>
<tr>
<td>Components</td>
<td>13</td>
</tr>
<tr>
<td>Functions</td>
<td>14</td>
</tr>
<tr>
<td>Screen Presentation</td>
<td>22</td>
</tr>
<tr>
<td>III. DEVELOPING THE FRAME-BASED MODEL</td>
<td>27</td>
</tr>
<tr>
<td>The Frame-based Model</td>
<td>27</td>
</tr>
<tr>
<td>Interaction with the Patient Database</td>
<td>34</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The Nursing Process</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>Development of the Nursing Care Plans: From the Systems Theory Point of View</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>The HELP Nursing Subsystem</td>
<td>10</td>
</tr>
<tr>
<td>4.</td>
<td>Five-step Process for Creating Nursing Care Plans</td>
<td>16</td>
</tr>
<tr>
<td>5.</td>
<td>The Actions Index for Problem 1</td>
<td>18</td>
</tr>
<tr>
<td>6.</td>
<td>A Review Screen for a Specific Care Plan</td>
<td>20</td>
</tr>
<tr>
<td>7.</td>
<td>Multiple Choice Window for Editing Outcomes</td>
<td>21</td>
</tr>
<tr>
<td>8.</td>
<td>Nursing Care Plan Menu Flow Diagram</td>
<td>24</td>
</tr>
<tr>
<td>9.</td>
<td>Nursing Care Plan Problem List Menu</td>
<td>25</td>
</tr>
<tr>
<td>10.</td>
<td>Nursing Care Plan Edit Option Menu</td>
<td>26</td>
</tr>
<tr>
<td>11.</td>
<td>Nursing Care Plan Problem Edit Menu</td>
<td>26</td>
</tr>
<tr>
<td>12.</td>
<td>The Frame-based Model for Computerized NCPs</td>
<td>28</td>
</tr>
<tr>
<td>13.</td>
<td>The Structure of the Main Program</td>
<td>30</td>
</tr>
<tr>
<td>14.</td>
<td>A Comparison of &quot;Delete String&quot; and &quot;Inactivate&quot; in Data Storage</td>
<td>32</td>
</tr>
</tbody>
</table>
15. Interaction between PAL and the Patient Database: Data Retrieval, Data Modification and Data Storage ...................... 35
17. The Search Criterion for the Review Frame .............................................. 39
18. The Specific Search Criterion for an Edit Frame ......................... 40
19. Purely Hierarchical PTXT Structure ................................................. 43
20. PTXT Constant Structure .................................................................. 43
21. An Example of the PTXT Constant Relation ................................. 44
22. HELP Decision Frame for Nursing Diagnosis............................ 48
23. Decision Support Uses in the Frame-based Model ...................... 49
24. CPU Time Recorded for Edit Frames ............................................... 52
25. Graphical Representation of the Timing Difference ..................... 56
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Timing Data for Edit Frames</td>
<td>53</td>
</tr>
<tr>
<td>2. Timing Comparision for Edit Frames</td>
<td>55</td>
</tr>
<tr>
<td>3. Suggested Nursing Diagnosis Data</td>
<td>58</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

I am grateful to Dr. T. Allan Pryor for his guidance and support as my advisor and the unique research experience with which he has provided me.

I also want to thank Dr. Peter J. Haug and Dr. Stan Huff for their helpful suggestions as my supervisory committee members.

I also want to express my appreciation to Ann Tinker, Dickey Johnson, Gary Melville, Marijo Burkes and Debbi Hinson for their help in the development of this project.

Acknowledgements are extended to the Department of Biophysics at LDS Hospital for providing the facilities required for this project.
CHAPTER I

INTRODUCTION

The Nursing Care Plan

The nursing care plan is an integral part of the nursing process defined by the American Nursing Association (ANA) and the Joint Commission on the Accreditation of Hospitals (JCAH) as assessment, planning, implementation and evaluation of nursing care (1). The nursing process is basically the scientific method of problem solving applied to nursing practice. It is used to identify patient problems, to plan and implement nursing care and to evaluate the results of care given. It is essential to the successful administration of nursing care. During this process, a nursing care plan is developed.

The nursing process is a continuing cycle, as shown in Figure 1. Assessment is the first phase of the nursing process and the basis of the nursing care plan. It includes patient data collection, patient data analysis and nursing diagnosis. A nursing diagnosis describes an active or potential health problem and all the possible causes related to that problem. The planning phase begins with the nursing diagnosis and ends with the development of a nursing care plan that is based on the nursing
diagnosis. Implementation is the actual giving of nursing care, which is defined by the nursing care plan. The final but continuous phase of the nursing process is evaluation of the given care. Evaluation indicates which problems have been solved and which need to be reassessed and replanned.

The nursing care plan, which is a practical way of putting the nursing process to work, would generally include the patient's problems, nursing care outcomes and actions. It is developed for a specific patient and should not be so general that it could be applied to everyone with the same medical diagnosis. A nursing diagnosis is not the same as a medical diagnosis. The "cause" portion of a nursing diagnosis is usually not a medical diagnosis and is intended to give a direction to nursing actions. Some medical diagnoses as "causes," however, are of limited value in directing nursing care. Individuals react differently to illness and the care plan should reflect this individuality. The care plan gives a guide to the adequate flow of nursing care throughout the patient's stages of illness.

It is quite clear that the development of nursing care plans is closely related to the nursing process. From the systems theory point of view, the nursing process can be characterized as a closed-loop system as shown in Figure 2. The nursing care plan is the output of the processor that includes assessment and planning while the input is patient data that are collected through assessment. The nursing care plan is then reintroduced as the input through a feedback mechanism that consists of
Figure 1. The Nursing Process.

Figure 2. Development of the Nursing Care Plan: From the Systems Theory Point of View.
implementation followed by evaluation for judging the results of nursing care. In this way, nursing care can be improved by comparing the outcomes of the care given with the expected outcomes. This analogy could be helpful for us to understand the generation of nursing care plans from an engineering point of view and also be beneficial to the automation of nursing care plans.

As we can see, the nursing care plan is the key component of the nursing process. Clinical use of the nursing care plan has been recognized as an important aid for planning and providing quality patient care. The value of the nursing care plan has been reinforced by the JCAH, which requires that nursing care plans be used before a hospital can be accredited.

**Computerization**

Although the theory of the nursing care plan is scientifically sound, manual attempts to implement the nursing care plan have been generally unsuccessful (2-4). With the manual method, development of nursing care plans has met with low nursing compliance in fulfilling the JCAH requirement that each hospitalized patient have a current, active nursing care plan (5). The written care plan is often incomplete and outdated. It is found to be only a "token" of accepting the care plan concept.

Because of the unsuccessful use of nursing care plans, a lot of important patient care data could be lost, which would otherwise be very useful in patient care, nursing education and research.
Any theory cannot be successfully translated into practice without an effective implementation method, so there is a need to find a powerful tool to assist in the implementation.

With the development of computer applications in hospitals, the health care industry has been changed dramatically. Many procedures in hospitals have been computerized with the introduction of Hospital Information Systems. The development of a Nursing Information System as a part of an integrated Hospital Information System has also become a necessity because of the increasing complexity of nursing care. The computer revolution has allowed nursing care to be conducted in a more efficient and sophisticated manner. As an integral part of the Nursing Information System, computerized nursing care plans have been developed and used in more and more hospitals (6-8). It is evident that computerization has provided an effective method of putting the nursing care plan into practice. It offers as much structure and flexibility as required.

**The HELP System**

The HELP (Health Evaluation through Logical Processing) system is a comprehensive hospital information system for acquiring medical data and implementing decision support which has been developed for over 20 years at the LDS Hospital in Salt Lake City, Utah. It currently operates on a 10-processor Non-stop Tandem System with more than 600 terminals and 100 printers at LDS Hospital, which is a 550-bed tertiary care hospital serving
the needs of the intermountain west.

The HELP system was designed to not only meet the needs of administrative, clinical and financial services, but also to provide decision-making capabilities. As one of the major objectives in the development of the HELP system, the system should be able to process medical decision logic (9). The decision logic should be modular, with control and interaction of modules determined by medical experts. Each decision module should be readily added to, modified and deleted so that the system is flexible and able to accommodate the changes of the patient database. Several unique tools were developed to fulfill this goal on the HELP system.

The PTXT Dictionary

The data dictionary is the "heart and soul" of a hospital information system and the foundation of all decision support methodologies (10). The PTXT (Pointer-to-TEXT) dictionary is the data dictionary on the HELP system. In any hospital information system, a computer needs access to a large amount of clinical data and medical knowledge in a form it can use. It is both easier and more economical for a computer to handle encoded data rather than natural text. PTXT is actually a data dictionary that links the English words and phrases along with their values to specific codes. These codes are used to store descriptions of patient care and medical knowledge rather than the actual text. Each data element is distinguished by a unique 8-byte code. The PTXT dictionary is used to encode descriptions
for computer storage and decode them for human understanding. The PTXT dictionary is organized in a hierarchical order from general to specific. The hierarchical structure allows a lower level item to "inherit" properties of the higher, more general classification to which it belongs without having to duplicate the information at each level. There are four different kinds of record structures used in the PTXT dictionary (11). Type 0 records are used for the fixed-size items. Type 1 records are used for the items with variable size and structure. Type 3 records are used in the decision-making logic of the HELP system. Type 7 records are used in transaction log file to assist in hospital information management functions.

Most of the data items are coded in Type 1. Type 1 data are used for the variable kinds of data, which can vary in both number of components and form. The 8-bytes, which represent the hierarchical levels, are data class, data type, field code, level, noun, adjective, adverb and modifier. The highest level is known as data class, which corresponds to different types of patient data, e.g., Data Class 28 represents nursing data and Data Class 8 represents pharmacy data. Within each data class, there are field codes, each of which are subdivided into nouns, adjectives and adverbs. A level indicator is included in the code to specify the deepest level in the hierarchy. Another component of Type 1 data is the modifier field. There are two kinds of modifiers: data class modifiers (DCM) and field code modifiers (FCM). Data class modifiers can be used with any item in the data class. Field code
The PTXT Application Language

The PTXT Application Language (PAL) was originally designed as a report generator and has become the HELP system's database language that provides a relational view of the PTXT dictionary. PAL is a high level modular programming language. Many application programs on the HELP system have been written in PAL because of its powerful capability of manipulating the patient database and handling screen interaction. The new version of PAL (PAL 18), which is still at the stage of development, is enhanced with numerous new features for accessing the patient database, using modern screen presentation technology for both display of results and acquisition of information from a terminal and using modular frame-based applications. A Tandem emulation program for PCs called PCTSCN has been developed. PCTSCN has incorporated a large library of C routines for menu and window manipulation. These C routines have been integrated into PAL 18.

The HELP Frame Language

The HELP frame language was designed as a special purpose interpreter to use the expert knowledge stored in a modular knowledge base. The knowledge is represented by frames that may be data acquisition, diagnosis, management, etc. Each frame is made up of various required and optional slots (12). All of the

modifiers are specific to a certain field code within a data class.
HELP frames are compiled into PAL code before being executed on the HELP system. Thus, medical knowledge can be represented by using the HELP frame language, which can be easily accepted by nurses and physicians who may not be familiar with computer programming.

**The HELP Nursing Subsystem**

A Nursing Information System can be defined as a computer system that collects, stores, processes, retrieves, displays and communicates timely information needed to do the following: 1) administer the nursing services and resources in a health care facility; 2) manage standardized patient care information for the delivery of nursing care; and 3) link the research resources and educational applications to nursing practice (13).

The Nursing Information System, a subsystem of HELP, has been under development for the last several years at LDS Hospital (14). The goals of the Nursing Information System are to provide an efficient means of nursing care and to promote the nursing process through computerization. The Nursing Information System will have four integral parts including nursing care plans, charting, staffing and HELP functions as shown in Figure 3. By computerizing and intercorrelating these four integral parts, many of their functions can be automated. The nurse will be required to spend less time interpreting and recording patient data. The steps of the nursing process required by JCAH will be followed.
Figure 3. The HELP Nursing Subsystem.
The Research Goals

The computerized nursing care plan is one of the major parts of the HELP's nursing subsystem (15). The trade-off between the speed and structure of a computerized model is usually one of the key problems in developing hospital information systems. Based on the different ways of interactions with the patient database, and the different structures of the PTXT dictionary and the patient database, there could be several different methods of implementing the computerized nursing care plan on the HELP system.

From the system point of view, the nursing care plan is one of the decision support applications on the HELP system (16) and it will be incorporated with other nursing applications such as nurse charting. Therefore, whether or not the nursing care plan is implemented efficiently will not only affect the nursing care plan itself but also the whole nursing information system. Besides, different hospitals could have different versions of the nursing care plan database, and the nursing care plan database could also be updated in the same hospital. It is clear that an effective and flexible model must be developed to implement the nursing care plan.

The goals of this research project are to provide:
1) A flexible model of computerization for accommodating the change of the nursing care plan database.
2) An effective method of programming on the basis of the trade-off between speed and structure.
3) The modern window features for the nursing care plan on the HELP system.

4) A basis for automating the nursing care plan and the nursing process on the HELP system.

In this research project, a frame-based model has been developed and different methods have been compared in terms of the speed and structure. As a initial study of automating the nursing care plans, HELP decision frames have been designed and integrated into the frame-based model for suggesting the nursing diagnoses.
CHAPTER II

COMPUTERIZED NURSING CARE PLANS

Components

A nursing care plan usually consists of several problems which are identified by nursing diagnosis. For each problem, there are causes related to that problem, expected outcomes and nursing actions.

The nursing care plan database used in this project was developed at the LDS Hospital and is currently being used on the HELP system.

The current nursing care plan on the HELP system was developed using a systems approach to patient problems. It consists of 16 problems that encompass the common abnormalities seen in acutely ill patients (7). These problems are:

1) Alteration in respiratory function;
2) Alteration in perfusion/cardiac output;
3) Alteration in fluid/electrolyte balance;
4) Alteration in neurological status;
5) Alteration in cardiac rhythm;
6) Alteration in patient comfort;
7) Alteration in elimination: bowel and bladder;
8) Infection/Wound management;
9) Impaired physical mobility;
10) Alteration in psychosocial states;
11) Alteration in nutrition;
12) Impaired coagulation status;
13) Knowledge deficit;
14) Self-care deficit/rehabilitation;
15) Cognitive-perceptual deficit; and
16) Female reproduction.

The causes related to each problem were included to assist the nursing diagnosis. Associated expected outcomes and nursing actions were determined for each problem.

**Functions**

A computerized nursing care plan should be able to support two main functions: 1) Creating an initial care plan; and 2) Editing an existing care plan. A patient identification number or name should be entered first to either create an initial care plan or edit an existing care plan for a specific patient. The computerized nursing care plan is designed to allow the nurse to create or edit one problem of the care plan at a time. On the problem list menu, the nurse can choose to create a new problem, to review and edit an existing problem, to print out the care plan, to inactivate the care plan, or to switch to another patient.
Create an Initial Care Plan

The creation of a new care plan is a five-step process as shown in Figure 4. The first step begins with the identification of specific patient problems. A problem list menu will be displayed on the screen to allow the nurse to choose from the 16 problems. Upon display of the problem list menu, a specific problem such as ALTERATION IN RESPIRATORY FUNCTION can be selected. The problem selected is considered to be an active problem unless it is flagged as a potential problem later by the nurse. The current problems are organized according to an anatomical systems approach. A revision will be made from the systems orientation to nursing diagnosis for problem identification.

The second step in creating the care plan is the selection of the causes related to the problem identified in step 1.

The third step is the identification of the expected outcomes for the patient. These goals are selected from a list of possible patient outcomes generated for each specific patient. The nurse may select all, some or none of the expected outcomes. If the nurse does not find an appropriate expected outcome in the list, one may be entered as free text to fit the needs of the specific patient.

The fourth step is the selection of nursing action categories to be addressed for the identified problem. These categories of nursing actions are chosen from the actions index. The actions index was developed as a menu of nursing actions commonly
Figure 4. Five-step Process for Creating Nursing Care Plans.
addressed under a specific patient problem. It allows the nurse to identify the specific needs of the patient without reviewing actions not applicable to that patient. Figure 5 presents the actions index for problem 1: Alteration in respiratory function. It allows the nurse to select actions relevant to oxygen therapy without reviewing actions for ventilators. Once the nurse has selected an item listed under the actions index, the standards of care will be automatically selected for that action. The standards of care are to provide the minimal safe care for a patient given the identified problem.

The final step is the selection of optional nursing actions necessary for the specific patient. Each actions index consists of two lists of actions. One is a list of standards of care, which is selected automatically as long as the actions index has been chosen for the patient. The other list is of optional actions which allow the nurse to individualize the care plan to the specific patient by selecting some or all of the actions. If the nurse chooses an action that has a numerical value associated with it, the specific values will be requested for that particular patient. In addition to the numerical value, free text entries may be selected to further individualize the care plans.

After this process has been completed for each identified patient problem, a new problem is created. This process can be repeated to create other problems identified by the nurse for the care plans.
1. ASSESSMENT AND NURSING CARE
2. ARTIFICIAL AIRWAY/SECRETION MANAGEMENT
3. TRACHEOSTOMY/SECRETION MANAGEMENT
4. NATURAL AIRWAY/SECRETION MANAGEMENT
5. OXYGEN THERAPY
6. OXYGEN HOODS AND TENTS
7. CHEST TUBES
8. BRONCHODILATORS
9. VENTILATORS

Figure 5. Actions Index for Problem 1.
Edit an Existing Care Plan

After the nursing care plan has been created for a specific patient, it can be easily updated as the patient's condition changes. When an existing problem is selected upon the problem list menu, the selected problem along with the related causes, the expected outcomes and the nursing actions will be displayed on the screen for review. Figure 6 presents a review screen for an existing problem. The nurse can choose to edit that problem, delete strings, inactivate a resolved problem, change an active problem to potential or change a potential problem to active.

Edit the Existing Problem

For editing the problem, the nurse can choose to edit the causes, the expected outcomes and the actions. Each of these editing options has a multiple choice window that not only displays the items but also indicates which items have previously been selected as shown in Figure 7. An asterisk indicates a previously selected item. The nurse can select one or more of the items or deselect one or more existing items by simply entering the item number. The multiple choice window for editing the existing problem is basically the same as the one for creating a new problem.

Delete Strings

Sometimes the nurse may select items by mistake when creating or editing an existing problem, and may discover the
1. ALTERATION IN RESPIRATORY FUNCTION

<table>
<thead>
<tr>
<th>RELATED CAUSES:</th>
<th>ACTIONS:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ATELECTASIS</td>
<td>1. ASSESS RETRACTION/ACCESSORY MUSCLE USE</td>
</tr>
<tr>
<td>2. PULMONARY EDEMA</td>
<td>2. REPORT ABG VALUES ABNORMAL FOR PATIENT</td>
</tr>
<tr>
<td>3. PNEUMONIA</td>
<td>3. ASSESS SECRETIONS FOR CHARACTER &amp;</td>
</tr>
<tr>
<td>4. POST-ANESTHESIA</td>
<td>VOLUME Q SHIFT</td>
</tr>
<tr>
<td></td>
<td>4. 24-HR SPUTUM COLLECTION</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EXPECTED OUTCOMES:</th>
<th>STANDARDS OF CARES:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. RESPIRATION REGULAR/ UNLABORED</td>
<td>1. ASSESS ADEQUACY OF FLUID BALANCE</td>
</tr>
<tr>
<td>2. RESPIRATORY RATE WNL FOR PATIENT</td>
<td>2. ASSESS BREATH SOUNDS Q SHIFT &amp; PRN</td>
</tr>
<tr>
<td>3. ABGS WNL FOR PATIENT ON ROOM AIR OR SUPP OXYGEN</td>
<td>3. EVALUATION RESP PATTERN</td>
</tr>
<tr>
<td></td>
<td>4. REPORT S/S HYPOXIA</td>
</tr>
</tbody>
</table>

LAST NURSE TO UPDATE CARE PLAN: BURKES, MARIJO
TIME INITIATED: 09/25/88 08:01

Figure 6. A Review Screen for a Specific Care Plan.
OUTCOMES: ALTERATION IN RESPIRATORY FUNCTION

1. RESPIRATIONS REGULAR/UNLABORED.
2. RESPIRATORY RATE WNL FOR PATIENT.
3. SPONTANEOUS VENTILATION.
4. CHEST CLEAR PER AUSCULTATION.
5. CHEST EXPANSION SYMMETRICAL.
6. ABGS WNL FOR PATIENT ON ROOM AIR OR SUPP O2.
7. INDEPENDENTLY HANDLES SECRETIONS.
8. PERFORMS NECESSARY SELF-CARE ROUTINES.
9. COMMENT: (FREE TEXT ENTRY).

Figure 7. Multiple Choice Window for Editing Outcomes.
mistakes only after reviewing that problem. In such a case, the nurse can choose to use the "delete strings" function, which will simply delete those strings from the patient database so that those mistakes can no longer be found as a part of the nursing care plan.

Inactivate a Resolved Problem

This function is chosen when the problem has been solved for a specific patient who previously had the problem.

Change an Active Problem to Potential

This function is used when an active problem becomes a potential problem.

Change a Potential Problem to Active

This function is used when a potential problem becomes an active problem.

Screen Presentation

As hospital information systems continue to increase in sophistication and complexity, so do the interface requirements that support user interaction. The human/computer interaction used in this computerized nursing care plan is basically menu-based interaction (17). The menu-based interaction is initiated and controlled by the information system. The system presents the menus to the user and the user provides responses.
to those menus.

Figure 8 presents a menu flow diagram for the computerized nursing care plan. There are a problem list menu, an edit option menu, a problem edit menu and groups of multiple choice windows. The problem list menu as shown in Figure 9 presents the patient name and the patient identification number for the selected patient, a list of problems and other options. An asterisk indicates an active problem and a "P" indicates a potential problem. The nurse can select a problem by simply entering the problem number. As for the other options, the nurse can enter "S" for "switch to another patient", "P" for "print out the care plan", "I" for "inactivate the care plan" and "R" for "return to the main menu.

Figure 10 shows the edit option menu and Figure 11 presents the problem edit menu. Figure 7 gives an example of the multiple choice window for editing the expected outcomes in problem 1.
Figure 8. Nursing Care Plan Menu Flow Diagram.
Figure 9. Nursing Care Plan Problem List Menu.
*** EDIT MENU ***

1. EDIT THE PROBLEM.
2. DELETE STRINGS.
3. INACTIVATE A RESOLVED PROBLEM.
4. CHANGE A POTENTIAL PROBLEM TO REAL.
5. CHANGE A REAL PROBLEM TO POTENTIAL.
6. --- RETURN TO THE PROBLEM LIST MENU.

SELECT ONE OF THE ABOVE:  

Figure 10. Nursing Care Plan Edit Option Menu.

*** EDIT PROBLEM ***

1. EDIT CAUSES.
2. EDIT OUTCOMES.
3. EDIT ACTIONS.
4. --- RETURN TO THE PROBLEM LIST MENU.

SELECT ONE OF THE ABOVE: 

Figure 11. Nursing Care Plan Problem Edit Menu.
CHAPTER III

DEVELOPING THE FRAME-BASED MODEL

The Frame-based Model

The frame-based model is a new architecture for implementing a knowledge driven hospital information system such as the HELP system. In a frame-based application, all functional modules can be defined as separate frames. All the frames are compiled separately into a unified frame file, which is simply an object file library. This feature makes it very flexible to add a new function, to modify or to delete an old function.

On the HELP system, a frame in the unified frame file is identified by a unique frame number consisting of four subitems which are data class, field code, frame number and a zero. For example, a frame number (28 1 200 0) represents a unique frame in the unified frame file.

On the basis of the frame architecture, a frame-based model has been developed for computerizing nursing care plans on the HELP system as shown in Figure 12. PAL 18 was used as a programming language in developing this model. The basic model is composed of a main program, a review frame, an edit option frame, edit frames, the patient database and a knowledge base
Figure 12. The Frame-based Model for Computerized NCPs.
Each frame can be designed and compiled independently. The patient data needed in one frame can be obtained by directly searching through the patient database, and can also be passed in as parameters from the main program or another frame.

The functions of the main program include patient selection, presentation of the problem list menu and overall control of the whole program. Figure 13 illustrates the general structure of the main program. Within the main program, there are four subroutines which are: 1) select patient; 2) read patient; 3) problem list menu; and 4) old or new. The "select patient" subroutine will allow the nurse to choose a new patient or switch from one patient to another. The "read patient" subroutine retrieves the current nursing care plan data for the selected patient from the patient database so that the problems can be identified as active, potential or inactive. For a new patient, there is no existing nursing care plan and all of the problems are identified as inactive. The "problem list menu" subroutine presents a menu of the problems with the information about whether a problem is active, potential or inactive. The nurse can choose from the menu either to create a new problem or to review and edit an existing problem. The "old or new" subroutine will identify whether the selected problem is a new or an existing problem. If it is a new problem, the edit frames including the cause frame, the outcome frame and the action frame will be called in sequence for the nurse to create that new problem. If it
Figure 13. The Structure of the Main Program.
is an old problem, the review frame will be called for the nurse to go over that old problem on the screen.

The review frame is used to retrieve the information about the existing problem from the patient database and display the whole problem. This display would include problem causes, outcomes and actions, as well as the time when the problem was updated and the name of the nurse who last updated the problem on the screen for review purposes.

The edit option frame is to provide the user with a menu of edit options including: 1) Edit the problem; 2) Delete strings; 3) Inactivate a resolved problem; 4) Change a potential problem to real; and 5) Change a real problem to potential.

The "Edit the problem" function is subdivided into editing causes, editing outcomes and editing actions by calling the corresponding edit frames. The "delete strings" function is achieved by physically deleting the strings from the patient database and then storing the same deleted strings together with the phrase "deleted strings" in the patient database. To inactivate a resolved problem, the problem is stored with the string "This problem has been inactivated." Figure 14 gives a comparison of the two functions in terms of data storage. The difference is that the "delete strings" function will actually delete the strings from the patient database, while the "inactivate problem" function will not. For a potential problem, it is stored with the string "This is a potential problem" and it is similar to the "inactivate problem" function.
Figure 14. A Comparison of "Delete String" and "Inactivate" in Data Storage.
Each problem has a corresponding edit module which consists of three edit frames, i.e., the cause frame, the outcome frame and the action frame. Each of those frames is basically a multiple choice window that allows the nurse to edit the causes, the outcomes or the actions for the specific problem. Those edit frames can be used either to create a new problem or to edit an existing problem. The unique function of this multiple choice window is that it is able to present not only the items to be selected but also the information about which items have previously been selected.

The knowledge base editor in the frame-based model is basically a knowledge tool which incorporates the decision support features of the HELP system, the modern screen presentation, the flexibility of modern expert systems in processing the expert knowledge and the ease of patient database access. It is used for generating the expert knowledge in a basic knowledge representation structure on the HELP system, which is the HELP language frame. This knowledge base editor allows the user not only to create windows, but also to interface the fields of the windows with the patient database and define decision support logic. The output of the editor is the HELP frames. Since all the knowledge frames are first compiled into PAL before execution, the effective PAL programs for those knowledge frames need to be developed first. The design of the knowledge base editor is based on those PAL programs that correspond to the HELP knowledge frames.
Two of the key problems in this frame-based model are:
1) Interaction with the patient database; 2) Structure of the PTXT dictionary.

**Interaction with the Patient Database**

A database language must have the capability of data retrieval, data modification and data storage. As a database language for the HELP system, the PTXT Application Language (PAL) has its unique feature to interact with the patient database, as shown in Figure 15.

On the HELP system, retrieving data from the patient database means searching through variable length data strings based on a given search criterion. When matching items are found, a data transformation from such a variable length data string into a relation with a fixed defined structure is performed.

In a PAL program, a basic search criterion must be first defined in a PAL statement called "RELATION" before any search can be attempted. A "RELATION" consists of one or more "ITEMs." After the "RELATION" is defined, the search can be triggered by a PAL statement called "BUILD." To limit the basic search criteria, the "BUILD" statement can be extended by appending additional conditions which need to be satisfied in order to put an entry in the "RELATION." The function $\text{EXIST}()$ and $\text{COUNT}()$ are defined to check the existence of a "RELATION" and the number of entries created during the search. As the program searches through the patient's data strings, a match is defined to occur when a data
Figure 15. Interaction between PAL and Patient Database: Data Retrieval, Data Modification and Data storage.
The string contains at least one "ITEM" defined in the "RELATION." The modification and storage can be done by using the PAL statement "DELETESTRING" and "MOVE^CODES BUFFER" and "PACK^CODES" and "SAVE^CODES."

The nursing care plan is updated by the nurse as the patient's condition changes in the hospital. The care plan histories are stored in the patient data base sequentially based on time.

In the frame-based model of the nursing care plan, there is a need to interact with the patient database in order to retrieve, modify the old nursing care plans and store the new nursing care plan in the patient database.

One of the functions of the main program is to identify which problem is active, potential or inactive in the current nursing care plan for a particular patient so that the active and potential problems can be marked on the problem list menu. In order to do this, the program needs to search through the patient database to find the current care plan for the patient. This function is achieved by defining a "RELATION" called PROBLEM and a "BUILD," as shown in Figure 16.

In the patient database of nursing care plans, a problem can not only be stored as active but also be marked as inactive or potential or deleted-string. If the current problem is marked by "deleted strings," the same problem without "deleted strings" before the current problem should be found as the most current one. If a current problem is marked by "inactive," it means that this problem is inactivated at present and will be treated as a
RELATION PROBLEMS;
BEGIN
  ITEM TIME         STRINGTIME;
  ITEM PROBLEM     CODE (28 1 1/16 * * * * *);
  ITEM INACTIVE    CODE (28 1 0 4 * * * 1);
  ITEM DEL^STRING  CODE (28 1 0 4 * * * 11);
  ITEM POTENTIAL   CODE (28 1 0 4 * * * 6);
END;

BUILD PROBLEM
WHERE $COUNT\ (\text{PROBLEM}) \geq 1 \text{ AND}
  NOT $\text{EXIST}\ (\text{PROBLEM}[$COUNT(\text{PROBLEM})].\text{DEL^STRING})
FROM $\text{NOW} \text{ BACKTO} *;

Figure 16. The Search Criterion for Finding the Current Problems.
new problem later on. The search criterion in the main program is defined to find only the most current problems unmarked by "deleted strings" during the search. The problems that are found could be active, potential or inactive. Problems can then be identified as active, potential or inactive.

In the review frame, a generic search criterion is defined for the search of the selected problem. The search criterion should be general enough for all of the patient problems including the related causes, the expected outcomes, the optional actions and the standard actions. A relation named "FIELD^CODE" and a corresponding "BUILD" are defined as shown in Figure 17.

As for each edit frame, information must be provided about which items have previously been selected. One way to do this is to search the patient data base directly based on a specific search criteria within each edit frame. Figure 18 gives an example of defining a specific search criterion for one of the edit frames. In this way, the specific information needed in each edit frame can be obtained by simply making a specific search. This method can be named as a "searching" method. Another way to do this is to pass the information into each edit frame from the review frame in which a search has already been made. However, the search in the review frame is based on a generic search criterion that can be used to find all the problems. Even if the information is passed into each edit frame, it is still necessary to do some comparison between the generic "RELATION" and the specific
RELATION FIELD\^CODE;
BEGIN
ITEM TIME STRINGTIME;
ITEM PROBLEM CODE(28 1 1/16 * * * * *);
ITEM CAUSE CODE(28 1 1/16 2/3 1 * *) SCALE(*);
ITEM OUTCOME CODE(28 1 1/16 2/3 3 * *) SCALE(*);
ITEM ACTION CODE(28 1 1/16 3 2 *) SCALE(*);
ITEM STANDARD CODE(28 1 1/16 3 4 *) SCALE(*);
ITEM INIT\^ITEM CODE(28 1 0 4 * * *2);
ITEM INACTIVE CODE(28 1 0 4 * * * 6);
ITEM POTENTIAL CODE(28 1 0 4 * * * 6);
ITEM DEL\^STRING CODE(28 1 0 4 * * * 11);
END;

BUILD FIELD\^CODE
WHERE $\text{EXIST(FIELD\^CODE.PROBLEM)}$
AND $\text{COUNT(FIELD\^CODE)}=1$
AND $\text{FC(FIELD\^CODE.PROBLEM)}=1$
AND NOT $\text{EXIST(FIELD\^CODE.DEL\^STRING)}$
FROM $\text{NOW BACKTO *}$;

Figure 17. The Search Criterion for the Review Frame.
RELATION P1OUTCOMES;
BEGIN
  ITEM TIME STRINGTIME;
  ITEM 01 CODE(28 1 1 2 3 1 0 0);
  ITEM 02 CODE(28 1 1 2 3 6 0 0);
  ITEM 03 CODE(28 1 1 2 3 9 0 0);
  ITEM 04 CODE(28 1 1 2 3 11 0 0);
  ITEM 05 CODE(28 1 1 2 3 16 0 0);
  ITEM 06 CODE(28 1 1 2 3 21 0 0);
  ITEM 07 CODE(28 1 1 2 3 31 0 0);
  ITEM 08 CODE(28 1 1 2 3 36 0 0);
  ITEM 09 CODE(28 1 1 2 3 200 0 0);
  ITEM DEL^STRING CODE(28 1 0 4 0 0 0 11);
END;

BUILD P1OUTCOMES UPDATE
  WHERE $COUNT(P1OUTCOMES)=1
    AND NOT $EXIST(DEL^STRING)
  FROM $NOW BACKTO *;

Figure 18. The Specific Search Criterion for an Edit Frame.
"RELATION" to find the specific items. The second method is called a "passing" method.

The advantage of the "searching" method is that the structure of the edit frame is very simple so that it is possible to develop a general purpose knowledge base editor to generate this kind of frame. The idea of using a knowledge base editor to generate the HELP frames corresponds with a goal of the HELP system. On the other hand, the structure of the "passing" method is more complex. It is difficult to have a general purpose knowledge base editor to generate the frames. In terms of the speed, the "passing" method could be faster than the "searching" method because there is no need to make another database search in each edit frame.

Structure of the PTXT Dictionary

The other key problem in the framed-base model is the structure of the PTXT dictionary. In the old structure of the PTXT dictionary, which is a purely hierarchical structure, different parent nodes cannot share the same child node. In the old PTXT dictionary structure of nursing care plans as shown in Figure 19, the causes, outcomes and actions of problem 1 are the subitems under problem 1. Those items cannot be shared by other types of patient data. However, in many cases, different types of patient data can have the same item. For the purely hierarchical structure of the PTXT dictionary, the same item has to be duplicated under the different types of patient data. This
will bring a lot of redundancy to the PTXT dictionary with the increase of the coded data. This is a disadvantage of a purely hierarchical database.

Fortunately, PAL provides a relational view of the PTXT dictionary and the capability of using multiple data class data. With the introduction of the PTXT constant structure as shown in Figure 20, an item which belongs to one type of patient data can be shared by other types of patient data by simply defining that item as a PTXT constant. Figure 21 gives an example of a PTXT constant relation. The PTXT constant heading item CAUSES has the Data Class 201, while item C1, C2 and C3 have the Data Class 202. The item C1, C2 and C3 can be shared by the heading item CAUSES as well as other types of patient data. In this way, once an item has been defined in one type of patient data, it is no longer necessary to duplicate that item in other types of patient data. Thus, the redundancy in the PTXT dictionary can be reduced by using the PTXT constant model.

**Variations on the Frame-based Model**

In summary, the major difference among the different methods discussed above can be focused on the structure of the edit frames. By combining the "passing" method and the "searching" method with the purely hierarchical structure data dictionary and the PTXT constant structure data dictionary, we have developed several different versions of the edit frames as follows:
Figure 19. Purely Hierarchical PTXT Structure.

Figure 20. PTXT Constant Structure.
RELATION PROB1^CAUSES;
BEGIN
ITEM TIME STRINGTIME;
ITEM CAUSES CODE (201 1 1 4 11 0 0 1) PTXT^CONSTANT;
BEGIN
ITEM C1 CODE (202 1 1 2 4 1 0 0);
ITEM C2 CODE (202 1 1 2 9 1 0 0);
ITEM C3 CODE (202 1 1 2 8 1 0 0);
END;
END;

Figure 21. An Example of a PTXT Constant Relation.
1) the "passing" method and the purely hierarchical structure;
2) the "searching" method and the purely hierarchical structure;
3) the "searching" method and the PTXT constant structure;
4) the "searching" method and the combination of the purely hierarchical and the PTXT constant structure.

The examples of those frames are presented in Appendices A and B.

**Decision Support Uses on the Frame-based Model**

The decision support in a hospital information system has almost become a necessity. Decision support requirement has come not only from the medical staff, but from the administrative and nursing staff as well. One decision support use is management decision support, which is the automatic generation of action-oriented decisions designed to improve the system state. The computerized nursing care plan is one of the decision support uses on the HELP system.

The automation of the nursing care plan can be divided into two steps. The first step is the automatic generation of the nursing diagnoses for the care plan. The second step is the automatic generation of the expected outcomes and nursing actions.

The approach to the nursing care plan is in the process of being changed from one of anatomical systems to that of diagnostic categories grouped under Functional Health Patterns.
The nursing care plan based on nursing diagnoses that are derived from actual patient data are more likely to be individualized and focused on the nurse's ability to assist the patient. The automation of the nursing care plan can help the nurse identify nursing diagnoses and generate better care plans.

As an initial study of automating the nursing care plan, eleven HELP frames have been built to suggest the corresponding nursing diagnoses. Those nursing diagnoses are:

1) Thought Processes, Altered;
2) Bowel Elimination, Altered: Constipation;
3) Bowel Elimination, Altered: Diarrhea.
4) Urinary Elimination, Altered Patterns: Incontinence, Total;
5) Bowel Elimination, Altered: Incontinence;
6) Self-Care Deficit: Feeding;
7) Self-Care Deficit: Bathing/Hygiene;
8) Skin Integrity, Impaired; Potential;
9) Skin Integrity, Impaired; Actual;
10) Mobility, Impaired Physical; and
11) Activity Intolerance.

Each nursing diagnosis has a corresponding decision frame. A decision frame consists of two parts. The first part searches the patient database to get the patient assessment data that are necessary to identify a specific nursing diagnosis. The second part is a group of rules that will conclude true or false for the nursing diagnosis based on the patient assessment data. For each nursing diagnosis, the rules have been set up by nursing experts. For
example, the rule for the nursing diagnosis of "Bowel Elimination, Altered; Diarrhea" is that if one or more of the symptoms such as liquid, loose and mucoid stool is presented, the nursing diagnosis of "Diarrhea" is concluded as true. Figure 22 gives an example of a decision frame for a specific nursing diagnosis. In this way, the expert knowledge can be used to assist the nurses to make their decisions.

The decision frames can be integrated into the frame-based model as shown in Figure 23.
TITLE: NURSING^DIAGNOSIS7 [28 : 1 : 807];
MESSAGE: ;
AUTHOR: [-0 0 0 0 0 0 0 ~ PING WANG-];
MAINTENANCE: ;
FRAME TYPE: DIAGNOSIS;
DESTINATION: ;
VARIABLE DECLARATIONS: C AS AN EXPRESSION CONTAINING
C1 WHICH IS |-28 1 60 3 100 20 5 0~| OR
C2 WHICH IS |-28 1 60 3 100 20 40 0~| OR
C3 WHICH IS |-28 1 60 3 100 20 10 0~| OR
C4 WHICH IS |-28 1 60 3 100 30 15 0~| OR
C5 WHICH IS |-28 1 60 3 100 30 16 0~| OR
C6 WHICH IS |-28 1 60 3 100 30 25 0~| OR
C7 WHICH IS |-28 1 60 3 100 30 35 0~| OR
C8 WHICH IS |-28 1 60 3 100 30 40 0~ |
FROM NOW BACK TO 10 YEARS AGO;

LOGIC:
SEARCH C;

IF EXIST(C)
THEN CONCLUDE TRUE;
ELSE CONCLUDE FALSE;

Figure 22. HELP Decision Frame for Nursing Diagnosis
Figure 23. Decision Support Uses in the Frame-based Model.
CHAPTER IV

RESULTS, DISCUSSIONS AND FUTURE WORK

Comparison of Different Methods

One of the key problems in this frame-based model is the structure and speed of the edit frames. Different edit frames have been developed as mentioned in Chapter III. These edit frames are presented in Appendices A and B. Each edit frame consists of three functional parts, which are data acquisition, data display and data storage.

In terms of the data acquisition, there is a "searching" method versus a "passing" method as discussed previously. One of the standards used to evaluate these edit frames is that the edit frames should be simple and general enough in structure so that a general knowledge base editor can be developed to generate those frames. The advantage of using the "searching" method is that the structure of the edit frames is very simple and is good for a general knowledge base editor. On the other hand, the structure of the "passing method" is much more complex (see Appendix A); and thus it is difficult to have a general knowledge base editor to generate such frames. Therefore, it is clear that the "searching" method is a better
choice for a general knowledge base editor than the "passing" method in terms of structure.

In order to compare the different edit frames in terms of speed, timing experiments have been designed. For both the "passing" and "searching" methods, the experiments have been performed on the edit frames with different number of items in a RELATION. The number of the items used in the test ranged from 4 to 20. Each edit frame had also been tested with the different number of strings previously stored in the patient database and the number of strings ranged from 2 to 20. The CPU time of the edit frames is shown in Figure 24. The CPU time recorded here is different from the actual user time, but they are closely related. Each edit frame with the same number of items and strings was tested 10 times. The average CPU time and its standard deviation were calculated based on the 10 repeated executions of each edit frame. The results are shown in Table 1. For example, if String#/ Item# = 16/20, the edit frame had 20 items in the RELATION, and it was tested when there were 16 current strings previously stored in the patient database. T1 represents the average CPU time for the "passing" method and S1 represents the corresponding sample standard deviation. Similarly, T2 represents the average CPU time for the "searching" method with the purely hierarchical structure and S2 represents the corresponding sample standard deviation. T3 represents the average CPU time for the "searching" method with the PTXT constant structure and S3 represents the
Figure 24. CPU Time Recorded for Edit Frames.
## Table 1
Timing Data for Edit Frames

<table>
<thead>
<tr>
<th>String#/Item#</th>
<th>Passing with Pure Hierarchy</th>
<th>Searching with Pure Hierarchy</th>
<th>Searching with PTXT Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T1</td>
<td>S1</td>
<td>T2</td>
</tr>
<tr>
<td>20/20</td>
<td>0.392</td>
<td>0.0103</td>
<td>0.432</td>
</tr>
<tr>
<td>16/20</td>
<td>0.377</td>
<td>0.0106</td>
<td>0.416</td>
</tr>
<tr>
<td>12/20</td>
<td>0.359</td>
<td>0.0088</td>
<td>0.400</td>
</tr>
<tr>
<td>8/20</td>
<td>0.344</td>
<td>0.0070</td>
<td>0.388</td>
</tr>
<tr>
<td>4/20</td>
<td>0.331</td>
<td>0.0074</td>
<td>0.370</td>
</tr>
<tr>
<td>2/20</td>
<td>0.323</td>
<td>0.0082</td>
<td>0.366</td>
</tr>
<tr>
<td>20/16</td>
<td>0.360</td>
<td>0.0105</td>
<td>0.403</td>
</tr>
<tr>
<td>16/16</td>
<td>0.359</td>
<td>0.0110</td>
<td>0.399</td>
</tr>
<tr>
<td>12/16</td>
<td>0.345</td>
<td>0.0071</td>
<td>0.381</td>
</tr>
<tr>
<td>8/16</td>
<td>0.332</td>
<td>0.0063</td>
<td>0.368</td>
</tr>
<tr>
<td>4/16</td>
<td>0.315</td>
<td>0.0071</td>
<td>0.351</td>
</tr>
<tr>
<td>2/16</td>
<td>0.303</td>
<td>0.0048</td>
<td>0.341</td>
</tr>
<tr>
<td>20/12</td>
<td>0.317</td>
<td>0.0067</td>
<td>0.360</td>
</tr>
<tr>
<td>16/12</td>
<td>0.317</td>
<td>0.0067</td>
<td>0.356</td>
</tr>
<tr>
<td>12/12</td>
<td>0.316</td>
<td>0.0070</td>
<td>0.351</td>
</tr>
<tr>
<td>8/12</td>
<td>0.306</td>
<td>0.0070</td>
<td>0.341</td>
</tr>
<tr>
<td>4/12</td>
<td>0.291</td>
<td>0.0057</td>
<td>0.330</td>
</tr>
<tr>
<td>2/12</td>
<td>0.283</td>
<td>0.0048</td>
<td>0.315</td>
</tr>
<tr>
<td>20/8</td>
<td>0.299</td>
<td>0.0099</td>
<td>0.325</td>
</tr>
<tr>
<td>16/8</td>
<td>0.297</td>
<td>0.0048</td>
<td>0.322</td>
</tr>
<tr>
<td>12/8</td>
<td>0.297</td>
<td>0.0095</td>
<td>0.319</td>
</tr>
<tr>
<td>8/8</td>
<td>0.295</td>
<td>0.0085</td>
<td>0.315</td>
</tr>
<tr>
<td>4/8</td>
<td>0.273</td>
<td>0.0048</td>
<td>0.300</td>
</tr>
<tr>
<td>2/8</td>
<td>0.265</td>
<td>0.0053</td>
<td>0.290</td>
</tr>
<tr>
<td>20/4</td>
<td>0.266</td>
<td>0.0070</td>
<td>0.294</td>
</tr>
<tr>
<td>16/4</td>
<td>0.265</td>
<td>0.0071</td>
<td>0.284</td>
</tr>
<tr>
<td>12/4</td>
<td>0.264</td>
<td>0.0070</td>
<td>0.282</td>
</tr>
<tr>
<td>8/4</td>
<td>0.266</td>
<td>0.0070</td>
<td>0.280</td>
</tr>
<tr>
<td>4/4</td>
<td>0.262</td>
<td>0.0073</td>
<td>0.272</td>
</tr>
<tr>
<td>2/4</td>
<td>0.245</td>
<td>0.0053</td>
<td>0.270</td>
</tr>
</tbody>
</table>
corresponding sample standard deviation. Table 2 gives a comparison of the timing difference of those methods. D21 means the difference between T2 and T1, i.e., D21 = T2-T1. Similarly, D32 = T3-T2 and D31 = T3-T1. Figure 25 gives a graphical representation for the timing difference. As can be seen, the "passing" method was faster than other type of methods. The "searching" method with PTXT constant structure was slower than the "searching" method with non-PTXT constant structure. The largest timing difference was less than 0.5 seconds. The difference, while significant, may in fact be within the requirements for the operational speed of the system.

Basically, there are three versions of the edit frame for the "searching" method that can be used in the frame-based model. In terms of the structure of the PTXT dictionary, there are the purely hierarchical structure, the PTXT constant structure and a combination of the two. The purely hierarchical structure edit frame can be used in single type data applications because of its purely hierarchical structure. The PTXT constant structure and the combination structure edit frames can be used in the cases of multiple data class applications.

**Suggesting Nursing Diagnoses**

In order to evaluate the results of the nursing diagnoses suggested by the corresponding decision frames, 11 nursing diagnosis decision frames have been tested on numbers of patients so that each nursing diagnosis has 25 cases. The
Table 2
Timing Comparison for Edit Frames

<table>
<thead>
<tr>
<th>String#/Item#</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>D21</th>
<th>D32</th>
<th>D31</th>
</tr>
</thead>
<tbody>
<tr>
<td>20/20</td>
<td>0.392</td>
<td>0.432</td>
<td>0.828</td>
<td>0.040</td>
<td>0.396</td>
<td>0.436</td>
</tr>
<tr>
<td>16/20</td>
<td>0.377</td>
<td>0.416</td>
<td>0.762</td>
<td>0.039</td>
<td>0.346</td>
<td>0.385</td>
</tr>
<tr>
<td>12/20</td>
<td>0.359</td>
<td>0.400</td>
<td>0.706</td>
<td>0.041</td>
<td>0.306</td>
<td>0.347</td>
</tr>
<tr>
<td>8/20</td>
<td>0.344</td>
<td>0.388</td>
<td>0.624</td>
<td>0.044</td>
<td>0.236</td>
<td>0.280</td>
</tr>
<tr>
<td>4/20</td>
<td>0.331</td>
<td>0.370</td>
<td>0.566</td>
<td>0.039</td>
<td>0.196</td>
<td>0.235</td>
</tr>
<tr>
<td>2/20</td>
<td>0.323</td>
<td>0.366</td>
<td>0.510</td>
<td>0.043</td>
<td>0.144</td>
<td>0.187</td>
</tr>
<tr>
<td>20/16</td>
<td>0.360</td>
<td>0.403</td>
<td>0.654</td>
<td>0.043</td>
<td>0.251</td>
<td>0.294</td>
</tr>
<tr>
<td>16/16</td>
<td>0.359</td>
<td>0.399</td>
<td>0.626</td>
<td>0.040</td>
<td>0.227</td>
<td>0.267</td>
</tr>
<tr>
<td>12/16</td>
<td>0.345</td>
<td>0.381</td>
<td>0.594</td>
<td>0.036</td>
<td>0.213</td>
<td>0.249</td>
</tr>
<tr>
<td>8/16</td>
<td>0.332</td>
<td>0.368</td>
<td>0.552</td>
<td>0.036</td>
<td>0.184</td>
<td>0.220</td>
</tr>
<tr>
<td>4/16</td>
<td>0.315</td>
<td>0.351</td>
<td>0.517</td>
<td>0.036</td>
<td>0.166</td>
<td>0.202</td>
</tr>
<tr>
<td>2/16</td>
<td>0.303</td>
<td>0.341</td>
<td>0.492</td>
<td>0.038</td>
<td>0.151</td>
<td>0.194</td>
</tr>
<tr>
<td>20/12</td>
<td>0.317</td>
<td>0.360</td>
<td>0.569</td>
<td>0.043</td>
<td>0.209</td>
<td>0.252</td>
</tr>
<tr>
<td>16/12</td>
<td>0.317</td>
<td>0.356</td>
<td>0.546</td>
<td>0.039</td>
<td>0.190</td>
<td>0.229</td>
</tr>
<tr>
<td>12/12</td>
<td>0.316</td>
<td>0.351</td>
<td>0.525</td>
<td>0.035</td>
<td>0.174</td>
<td>0.209</td>
</tr>
<tr>
<td>8/12</td>
<td>0.306</td>
<td>0.341</td>
<td>0.485</td>
<td>0.035</td>
<td>0.144</td>
<td>0.179</td>
</tr>
<tr>
<td>4/12</td>
<td>0.291</td>
<td>0.330</td>
<td>0.461</td>
<td>0.039</td>
<td>0.131</td>
<td>0.170</td>
</tr>
<tr>
<td>2/12</td>
<td>0.283</td>
<td>0.315</td>
<td>0.443</td>
<td>0.032</td>
<td>0.128</td>
<td>0.160</td>
</tr>
<tr>
<td>20/8</td>
<td>0.299</td>
<td>0.325</td>
<td>0.509</td>
<td>0.026</td>
<td>0.184</td>
<td>0.210</td>
</tr>
<tr>
<td>16/8</td>
<td>0.297</td>
<td>0.322</td>
<td>0.483</td>
<td>0.025</td>
<td>0.161</td>
<td>0.186</td>
</tr>
<tr>
<td>12/8</td>
<td>0.297</td>
<td>0.319</td>
<td>0.457</td>
<td>0.022</td>
<td>0.138</td>
<td>0.160</td>
</tr>
<tr>
<td>8/8</td>
<td>0.295</td>
<td>0.315</td>
<td>0.422</td>
<td>0.020</td>
<td>0.107</td>
<td>0.127</td>
</tr>
<tr>
<td>4/8</td>
<td>0.273</td>
<td>0.300</td>
<td>0.381</td>
<td>0.027</td>
<td>0.081</td>
<td>0.108</td>
</tr>
<tr>
<td>2/8</td>
<td>0.265</td>
<td>0.290</td>
<td>0.355</td>
<td>0.025</td>
<td>0.065</td>
<td>0.090</td>
</tr>
<tr>
<td>20/4</td>
<td>0.266</td>
<td>0.294</td>
<td>0.393</td>
<td>0.028</td>
<td>0.096</td>
<td>0.124</td>
</tr>
<tr>
<td>16/4</td>
<td>0.265</td>
<td>0.284</td>
<td>0.379</td>
<td>0.019</td>
<td>0.095</td>
<td>0.114</td>
</tr>
<tr>
<td>12/4</td>
<td>0.264</td>
<td>0.282</td>
<td>0.364</td>
<td>0.018</td>
<td>0.082</td>
<td>0.100</td>
</tr>
<tr>
<td>8/4</td>
<td>0.266</td>
<td>0.280</td>
<td>0.351</td>
<td>0.014</td>
<td>0.071</td>
<td>0.085</td>
</tr>
<tr>
<td>4/4</td>
<td>0.262</td>
<td>0.272</td>
<td>0.331</td>
<td>0.010</td>
<td>0.059</td>
<td>0.069</td>
</tr>
<tr>
<td>2/4</td>
<td>0.245</td>
<td>0.270</td>
<td>0.308</td>
<td>0.025</td>
<td>0.038</td>
<td>0.063</td>
</tr>
</tbody>
</table>
Figure 25. Graphical Representation of the Timing Difference.
suggested nursing diagnosis for each case was recorded and compared to the actual nursing charting. The true positive rate (TPR), false positive rate (FPR), true negative rate (TNR) and false negative rate (FNR) were calculated for each nursing diagnosis as shown in Table 3. The true negative rate and false positive rate cannot be computed for nursing diagnosis No. 8 and 10 because the numbers of true negative and false positive were both zero. As can be seen, the false positive rates were relatively small for all the nursing diagnoses. It can be concluded that the probability of mistakenly suggesting a nursing diagnosis is small by using these decision frames for the nursing diagnoses.

The true positive rate for nursing diagnosis No. 4, which is "Urinary Elimination, Altered Patterns; Incontinence, Total," was only 0.444. The rule for this nursing diagnosis is that if the symptom of "Incontinent" and/or "Adult diaper change" presents, the nursing diagnosis is concluded as true. Some cases resulted from the identification item "Adult diaper change" as a predictor of urinary incontinence. However, the patient had a Foley urine catheter in place so there would not have been an adult diaper on the patient for the possible occurrence of urine incontinence. So the wrong suggestion was resulted from inaccurate rules.

For further improvement of the suggested nursing diagnoses, the rules for the nursing diagnoses can be modified based on the nurse knowledge and experience to achieve better results. This can be an ever-changing process as knowledge of the nursing
Table 3

Suggested Nursing Diagnosis Data

<table>
<thead>
<tr>
<th>Nursing Diagnosis No.</th>
<th>TPR</th>
<th>FNR</th>
<th>TNR</th>
<th>FPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.737</td>
<td>0.263</td>
<td>0.833</td>
<td>0.167</td>
</tr>
<tr>
<td>2</td>
<td>1.000</td>
<td>0.000</td>
<td>0.909</td>
<td>0.091</td>
</tr>
<tr>
<td>3</td>
<td>0.813</td>
<td>0.187</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>0.444</td>
<td>0.556</td>
<td>0.813</td>
<td>0.187</td>
</tr>
<tr>
<td>5</td>
<td>0.846</td>
<td>0.154</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>0.958</td>
<td>0.042</td>
<td>1.000</td>
<td>0.000</td>
</tr>
<tr>
<td>8</td>
<td>1.000</td>
<td>0.000</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>9</td>
<td>0.818</td>
<td>0.182</td>
<td>0.857</td>
<td>0.143</td>
</tr>
<tr>
<td>10</td>
<td>0.960</td>
<td>0.040</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>11</td>
<td>1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
diagnoses is improved or a new nursing diagnosis comes up. However, the basic structure of the decision frame will remain the same.

**Future Work and Conclusions**

The HELP system has its unique decision-making capabilities in addition to general administrative and clinical functions. One goal in developing the HELP's Nursing Information System is the automation of the nursing care plans. On the basis of patient data and other necessary information, the system will be able to automatically generate suggested nursing care plans, which include nursing diagnosis, expected outcomes and nursing actions for a particular patient so that the nursing care plans will be generated much more efficiently.

The frame-based design provides the flexibility of adding new functional modules, modifying the old functional modules and the foundation of automating the nursing care plan.

The frame-based model integrates the three major functions of any information system, which are data retrieve, data modification and data storage and also provides HELP decision support functions. It is necessary to point out that although the frame-based model along with the edit frames is developed for computerizing the nursing care plans in this project, it could be a general model for many other applications on the HELP system.
APPENDIX A

THE EDIT FRAME FOR "PASSING METHOD"
THE EDIT FRAME — "PASSING" METHOD

BEGIN

PARAMETER RELATION FIELD CODE;
BEGIN
ITEM TIME STRINGTIME;
ITEM INACTIVE CODE(28 1 0 4 * * * 1);
ITEM POT CODE(28 1 0 4 * * * 6);
ITEM PROB CODE(28 1 1/16 * * * * *);
ITEM CAUSE CODE(28 1 1/16 2/3 1 * *) SCALE(*);
ITEM OUTCOME CODE(28 1 1/16 2/3 3 * *) SCALE(*);
ITEM STANDARD CODE(28 1 1/16 3 4 *) SCALE(*);
ITEM ACTION CODE(28 1 1/16 3 2 *) SCALE(*);
ITEM INIT TIME CODE(28 1 0 4 * * * 2);
END;

PARAMETER VARIABLE NEW;
VARIABLE DROSSO, L, NUMBER OF OUTCOMES;
VARIABLE NULL CHAR[20], COUNT, NUMANS, SELECT CHAR[30],
ISSELECTED, FKEY, STAT WINDOW, P1OUTCOMES;

RELATION O1;
BEGIN
ITEM TIME STRINGTIME;
ITEM O1 CODE(28 1 1 2 3 1 0 0);
END;
RELATION O2;
BEGIN
ITEM TIME STRINGTIME;
ITEM O2 CODE(28 1 1 2 3 6 0 0);
END;
RELATION O3;
BEGIN
ITEM TIME STRINGTIME;
ITEM O3 CODE(28 1 1 2 3 9 0 0);
END;
RELATION O4;
BEGIN
ITEM TIME STRINGTIME;
ITEM O4 CODE(28 1 1 2 3 11 0 0);
END;
RELATION O5;
BEGIN
ITEM TIME STRINGTIME;
ITEM O5 CODE(28 1 1 2 3 16 0 0);
END;
RELATION 06;
BEGIN
  ITEM TIME STRINGTIME;
  ITEM 06 CODE(28 1 1 2 3 21 0 0);
END;
RELATION 07;
BEGIN
  ITEM TIME STRINGTIME;
  ITEM 07 CODE(28 1 1 2 3 31 0 0);
END;
RELATION 08;
BEGIN
  ITEM TIME STRINGTIME;
  ITEM 08 CODE(28 1 1 2 3 36 0 0);
END;
RELATION 09;
BEGIN
  ITEM TIME STRINGTIME;
  ITEM 09 CODE(28 1 1 2 3 200 0 0) FREETEXT(80);
END;

?NOLIST
?SOURCE "DATA PALVC LITERALS";

?LIST
ALLOCATE 01;
ALLOCATE 02;
ALLOCATE 03;
ALLOCATE 04;
ALLOCATE 05;
ALLOCATE 06;
ALLOCATE 07;
ALLOCATE 08;
ALLOCATE 09;
P1OUTCOMES:=MCNEW(2,4,0,23,"OUTCOMES IN ALTERATION IN RESPIRATORY FUNCTION",COOKED+CURSORS+CENTER+BORDER+BD1,SELECT);
IF NEW=1 THEN NUMBER OF OUTCOMES:=0
ELSE NUMBER OF OUTCOMES:=$COUNT(FIELD CODE OUTCOME);
L:=1;
ISSELECTED:=0;
IF L<NUMBER OF OUTCOMES THEN
BEGIN
  IF $ADJ(01.01)=$ADJ(FIELD CODE OUTCOME[L]) THEN
    BEGIN
      ISSELECTED:=MCSELECTED;
      L:=L+1;
    END;
  END;
MCITEM(P1OUTCOMES,1,1,0,"RESPIRATIONS REGULAR/UNLABERED",0.01.01, NULL,1,ISSELECTED);
ISSELECTED:=0;
IF L<=NUMBER"OF"OUTCOMES THEN
BEGIN
  IF $ADJ(O2.02)=$ADJ(FIELD"CODE.OUTCOME[L]) THEN
  BEGIN
    ISSELECTED:=MCSELECTED;
    L:=L+1;
  END;
END;
MCITEM(P1OUTCOMES,2,2,0,"RESPIRATORY RATE WNL FOR
PATIENT",0,02.02,NULL,1,ISSELECTED);
ISSELECTED:=0;
IF L<=NUMBER"OF"OUTCOMES THEN
BEGIN
  IF $ADJ(O3.03)=$ADJ(FIELD"CODE.OUTCOME[L]) THEN
  BEGIN
    ISSELECTED:=MCSELECTED;
    L:=L+1;
  END;
END;
MCITEM(P1OUTCOMES,3,3,0,"SPONTANEOUS VENTILATION",0,03.03,
NULL,1,ISSELECTED);
ISSELECTED:=0;
IF L<=NUMBER"OF"OUTCOMES THEN
BEGIN
  IF $ADJ(O4.04)=$ADJ(FIELD"CODE.OUTCOME[L]) THEN
  BEGIN
    ISSELECTED:=MCSELECTED;
    L:=L+1;
  END;
END;
MCITEM(P1OUTCOMES,4,4,0,"CHEST CLEAR PER AUSCULTATION",0,04.04,
NULL,1,ISSELECTED);
ISSELECTED:=0;
IF L<=NUMBER"OF"OUTCOMES THEN
BEGIN
  IF $ADJ(O5.05)=$ADJ(FIELD"CODE.OUTCOME[L]) THEN
  BEGIN
    ISSELECTED:=MCSELECTED;
    L:=L+1;
  END;
END;
MCITEM(P1OUTCOMES,5,5,0,"CHEST EXPANSION SYMMETRICAL",0,05.05,
NULL,1,ISSELECTED);
ISSELECTED:=0;
IF L<=NUMBER"OF"OUTCOMES THEN
BEGIN
  IF $ADJ(O6.06)=$ADJ(FIELD"CODE.OUTCOME[L]) THEN
  BEGIN
    ISSELECTED:=MCSELECTED;
    L:=L+1;
  END;
END;
MCITEM(P1OUTCOMES,6,6,0,"CHEST PAIN",0,06.06,NULL,1,ISSELECTED);
BEGIN
  ISSELECTED:=MCSELECTED;
  L:=L+1;
END;
END;
MCITEM(P1OUTCOMES,6,6,"ABGS MNL FOR PATIENT ON ROOM AIR OR SUPP O2",0.06.06,NULL,1,ISSELECTED);
ISSELECTED:=0;
IF L<=NUMBER*OF*OUTCOMES THEN
BEGIN
  IF $ADJ(07.07)=$ADJ(FIELD^CODE.OUTCOME[L]) THEN
    BEGIN
      ISSELECTED:=MCSELECTED;
      L:=L+1;
    END;
  END;
MCITEM(P1OUTCOMES,7,7,"INDEPENDENTLY HANDLES SELF-CARE ROUTINES",0.07.07,NULL,1,ISSELECTED);
ISSELECTED:=0;
IF L<=NUMBER*OF*OUTCOMES THEN
BEGIN
  IF $ADJ(08.08)=$ADJ(FIELD^CODE.OUTCOME[L]) THEN
    BEGIN
      ISSELECTED:=MCSELECTED;
      L:=L+1;
    END;
  END;
MCITEM(P1OUTCOMES,8,8,"PERFORMS NECESSARY SELF-CARE ROUTINES",0.08.08,NULL,1,ISSELECTED);
ISSELECTED:=0;
IF L<=NUMBER*OF*OUTCOMES THEN
BEGIN
  IF $ADJ(09.09)=$ADJ(FIELD^CODE.OUTCOME[L]) THEN
    BEGIN
      ISSELECTED:=MCSELECTED;
      L:=L+1;
    END;
  END;
MCITEM(P1OUTCOMES,9,9,"COMMENT:(FREE TEXT ENTRY)",0.09.09, NULL,1,ISSELECTED);
FKEY:=KEYSET(ESCAPE,CSF4,CSF10);
DROSS0:=FMCREAD(FKEY,0,P1OUTCOMES,NULL,NUM^ANS);
IF DROSS0<>13 THEN RETURN DROSS0;
MCHIDE;
IF $VAL(SELECT[1,1]) THEN MOVE^CODES BUFFER 01.01;
IF $VAL(SELECT[1,2]) THEN MOVE^CODES BUFFER 02.02;
IF $VAL(SELECT[1,3]) THEN MOVE^CODES BUFFER O3.03;
IF $VAL(SELECT[1,4]) THEN MOVE^CODES BUFFER O4.04;
IF $VAL(SELECT[1,5]) THEN MOVE^CODES BUFFER O5.05;
IF $VAL(SELECT[1,6]) THEN MOVE^CODES BUFFER O6.06;
IF $VAL(SELECT[1,7]) THEN MOVE^CODES BUFFER O7.07;
IF $VAL(SELECT[1,8]) THEN MOVE^CODES BUFFER O8.08;
IF $VAL(SELECT[1,9]) THEN MOVE^CODES BUFFER O9.09;
END;
APPENDIX B

THE EDIT FRAMES FOR "SEARCHING METHOD"
THE EDIT FRAME — "SEARCHING" METHOD
NON-PTXT CONSTANT MODEL

?FREETEXT 100
SECTION P1OUTCOME (28 1 303 0);
BEGIN
  VARIABLE DROSS0,L,NUMBER*OF*OUTCOMES;
  VARIABLE NULL CHAR[20], COUNT,NUM^ANS,SELECT CHAR[30],
  ISSELECTED,FKEY,STAT*WINDOW,P1OUTCOMES;
  RELATION PROB1^OUTCOMES;
BEGIN
  ITEM TIME STRINGTIME;
  ITEM 01 CODE(28 1 2 3 1 0 0);
  ITEM 02 CODE(28 1 2 3 6 0 0);
  ITEM 03 -CODE(28 1 2 3 9 0 0);
  ITEM 04 CODE(28 1 2 3 1 1 0 0);
  ITEM 05 CODE(28 1 2 3 1 6 0 0);
  ITEM 06 CODE(28 1 2 3 2 1 0 0);
  ITEM 07 CODE(28 1 2 3 3 1 0 0);
  ITEM 08 CODE(28 1 2 3 3 6 0 0);
  ITEM 09 CODE(28 1 2 3 2 0 0 0) FREETEXT(80);
  ITEM DEL^STRING CODE(28 1 0 4 0 0 0 11);
END;
?NOLIST
?SOURCE "DATA PALVC LITERALS";
?LIST

*** SPECIFIC "BUILD" TO FIND THE OUTCOMES ***

BUILD PROB1^OUTCOMES FOR^UPDATE
  WHERE $COUNT(PROB1^OUTCOMES)=1
  AND NOT $EXIST(PROB1^OUTCOMES.DEL^STRING)
  FROM $NOW BACK TO *;

*** MULTIPLE CHOICE WINDOW FOR EDITING THE OUTCOMES ***

P1OUTCOMES:=MCNEW(2,4,0,9,"OUTCOMES IN ALTERATION IN RESPIRATORY
  FUNCTION",COOKED+CURSORS+CENTER+BORDER+BD1,SELECT);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.O1) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,1,1,0,"RESPIRATIONS REGULAR/UNLABERED",0,
  PROB1^OUTCOMES.O1,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.O2) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,2,2,0,"RESPIRATORY RATE WNL FOR PATIENT",0,
PROB1^OUTCOMES.02,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.03) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,3,3,0,"SPONTANEOUS VENTILATION",0,
PROB1^OUTCOMES.03,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.04) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,4,4,0,"CHEST CLEAR PER AUSCULTATION",0,
PROB1^OUTCOMES.04,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.05) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,5,5,0,"CHEST EXPANSION SYMMETRICAL",0,
PROB1^OUTCOMES.05,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.06) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,6,6,0,"ABGS MNL FOR PATIENT ON ROOM AIR OR SUPP O2",0,
PROB1^OUTCOMES.06,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.07) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,7,7,0,"INDEPENDENTLY HANDLES SELF-CARE ROUTINES",0,
PROB1^OUTCOMES.07,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.08) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,8,8,0,"PERFORMS NECESSARY SELF-CARE ROUTINES",0,
PROB1^OUTCOMES.08,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.09) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,9,9,0,"COMMENT:(FREE TEXT ENTRY)",0,
PROB1^OUTCOMES.09,NULL,1,ISSELECTED);

FKEY:=KEYSET(ESCAPE,CSF4,CSF10);
DROSS0:=FMCREAD(FKEY,0,P1OUTCOMES,NULL,NUM^ANS);
IF DROSS0<>13 THEN RETURN DROSS0;
MCHIDE;
IF $VAL(SELECT[1,1]) THEN MOVE^CODES BUFFER PROB1^OUTCOMES.O1;
IF $VAL(SELECT[1,2]) THEN MOVE^CODES BUFFER PROB1^OUTCOMES.O2;
IF $VAL(SELECT[1,3]) THEN MOVE^CODES BUFFER PROB1^OUTCOMES.O3;
IF $VAL(SELECT[1,4]) THEN MOVE^CODES BUFFER PROB1^OUTCOMES.O4;
IF $VAL(SELECT[1,5]) THEN MOVE^CODES BUFFER PROB1^OUTCOMES.O5;
IF $VAL(SELECT[1,6]) THEN MOVE^CODES BUFFER PROB1^OUTCOMES.O6;
IF $VAL(SELECT[1,7]) THEN MOVE^CODES BUFFER PROB1^OUTCOMES.O7;
IF $VAL(SELECT[1,8]) THEN MOVE^CODES BUFFER PROB1^OUTCOMES.O8;
IF $VAL(SELECT[1,9]) THEN MOVE^CODES BUFFER PROB1^OUTCOMES.O9;
END;
THE EDIT FRAME --- "SEARCHING" METHOD
PTXT CONSTANT MODEL

FREETEXT 100
SECTION P1OUTCOME (28 1 403 0);
BEGIN

VARIABLE DROSO, L, NUMBER OF OUTCOMES;
VARIABLE NULL CHAR[20], COUNT, NUM^ANS, SELECT CHAR[30],
ISSELECTED, FKEY, STAT^WINDOW, P1OUTCOMES;

RELATION PROB1*OUTCOMES;
BEGIN

ITEM TIME STRINGTIME;
ITEM OUTCOMES CODE(201 1 4 0 0 0 1) PTXT^CONSTANT;
BEGIN

ITEM O1 CODE(28 1 1 2 3 1 0 0);
ITEM O2 CODE(28 1 1 2 3 6 0 0);
ITEM O3 CODE(28 1 1 2 3 9 0 0);
ITEM O4 CODE(28 1 1 2 3 11 0 0);
ITEM O5 CODE(28 1 1 2 3 16 0 0);
ITEM O6 CODE(28 1 1 2 3 21 0 0);
ITEM O7 CODE(28 1 1 2 3 31 0 0);
ITEM O8 CODE(28 1 1 2 3 36 0 0);
ITEM O9 CODE(28 1 1 2 3 200 0 0) FREETEXT(80);

END;
ITEM DEL^STRING CODE(201 1 1 0 4 0 0 0 11);
END;

NOLIST
SOURCE "DATA PALVC LITERALS";

LIST

SPECIFIC "BUILD" TO FIND THE OUTCOMES

BUILD PROB1*OUTCOMES FOR UPDATE
WHERE $COUNT(PROB1*OUTCOMES)=1
AND NOT $EXIST(PROB1*OUTCOMES.DEL^STRING)
FROM $NOW BACK TO **;

MULTIPLE CHOICE WINDOW FOR EDITING THE OUTCOMES

P1OUTCOMES=MCNEW(2,4,0,9,"OUTCOMES IN ALTERATION IN RESPIRATORY FUNCTION",COOKED+CURSORS+CENTER+BD1,SELECT);

ISSELECTED=0;
IF $EXIST(PROB1*OUTCOMES.OUTCOMES.01) THEN ISSELECTED=MCSELECTED;
MCITEM(P1OUTCOMES,1,1,0,"RESPIRATIONS REGULAR/UNLABERED",0,
PROB1*OUTCOMES.OUTCOMES.01,NULL,1,ISSELECTED);
ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.OUTCOMES.02) THEN ISSELECTED:=MCSELECTED;
MCITEM(P10UTCOMES,2,2,0,"RESPIRATORY RATE WNL FOR PATIENT",0,
PROB1^OUTCOMES.OUTCOMES.02,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.OUTCOMES.03) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,3,3,0,"SPONTANEOUS VENTILATION",0,
PROB1^OUTCOMES.OUTCOMES.03,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.OUTCOMES.04) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,4,4,0,"CHEST CLEAR PER AUSCULTATION",0,
PROB1^OUTCOMES.OUTCOMES.04,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.OUTCOMES.05) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,5,5,0,"CHEST EXPANSION SYMMETRICAL",0,
PROB1^OUTCOMES.OUTCOMES.05,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.OUTCOMES.06) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,6,6,0,"ABGS WNL FOR PATIENT ON ROOM AIR OR SUPP O2",0,
PROB1^OUTCOMES.OUTCOMES.06,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.OUTCOMES.07) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,7,7,0,"INDEPENDENTLY HANDLES SELF-CARE ROUTINES",0,
PROB1^OUTCOMES.OUTCOMES.07,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.OUTCOMES.08) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,8,8,0,"PERFORMS NECESSARY SELF-CARE ROUTINES",0,
PROB1^OUTCOMES.OUTCOMES.08,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.OUTCOMES.09) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,9,9,0,"COMMENT:(FREE TEXT ENTRY)",0,
PROB1^OUTCOMES.OUTCOMES.09,NULL,1,ISSELECTED);

FKEY:=KEYSET(ESCAPE,CSF4,CSF10);
DROSSO:=FMCREAD(FKEY,0,P1OUTCOMES,NULL,NUM^ANS);
IF DROSSO<>13 THEN RETURN DROSSO;
MCHIDE;
IF NOT $VAL(SELECT[1,1])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.O1) := FALSE;
IF NOT $VAL(SELECT[1,2])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.O2) := FALSE;
IF NOT $VAL(SELECT[1,3])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.O3) := FALSE;
IF NOT $VAL(SELECT[1,4])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.O4) := FALSE;
IF NOT $VAL(SELECT[1,5])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.O5) := FALSE;
IF NOT $VAL(SELECT[1,6])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.O6) := FALSE;
IF NOT $VAL(SELECT[1,7])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.O7) := FALSE;
IF NOT $VAL(SELECT[1,8])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.O8) := FALSE;
IF NOT $VAL(SELECT[1,9])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.O9) := FALSE;
MOVE^CODES BUFFER PROB1^OUTCOMES.OUTCOMES;
END;
THE EDIT FRAME --- "SEARCHING" METHOD

COMBINATION MODEL

?FREETEXT 100
SECTION P1OUTCOME (28 1 503 0);
BEGIN
VARIABLE DRSS0, L, NUMBER OF OUTCOMES;
VARIABLE NULL CHAR[20], COUNT, NUM^ANS, SELECT CHAR[30],
ISSELECTED, FKEY, STAT^WINDOW, P1OUTCOMES;
RELATION PROB1^OUTCOMES;
BEGIN
ITEM TIME STRINGTIME;
ITEM OUTCOMES CODE(201 1 1 4 0 0 0 1) PTXT^CONSTANT;
BEGIN
ITEM O1 CODE(28 1 1 2 3 1 0 0);
ITEM O2 CODE(28 1 1 2 3 6 0 0);
ITEM O3 CODE(28 1 1 2 3 9 0 0);
ITEM O4 CODE(28 1 1 2 3 1 1 0 0);
ITEM O5 CODE(28 1 1 2 3 1 6 0 0);
END;
ITEM O6 CODE(201 1 1 2 3 21 0 0);
ITEM O7 CODE(201 1 1 2 3 31 0 0);
ITEM O8 CODE(201 1 1 2 3 36 0 0);
ITEM O9 CODE(201 1 1 2 3 200 0 0)
FREETEXT(80);
ITEM DEL^STRING CODE(201 1 1 0 4 0 0 0 11);
END;
?NOLIST
?SOURCE "DATA PALVC LITERALS";
?LIST

*** SPECIFIC "BUILD" TO FIND THE OUTCOMES ***

BUILD PROB1^OUTCOMES FOR UPDATE
WHERE $COUNT(PROB1^OUTCOMES)=1
AND NOT $EXIST(PROB1^OUTCOMES.DEL^STRING)
FROM $NOW BACK TO *;

*** MULTIPLE CHOICE WINDOW FOR EDITING THE OUTCOMES ***

P1OUTCOMES:=MCNEW(2,4,0,9,"OUTCOMES IN ALTERATION IN RESPIRATORY FUNCTION",COOKED+CURSORS+CENTER+BD1,SELECT);
ISSELECTED:=0;
IF $EXIST(PROB1^OUTCOMES.OUTCOMES.01) THEN ISSELECTED:=MCSELECTED;
MCITEM(P1OUTCOMES,1,1,0,"RESPIRATIONS REGULAR/UNLABERED",0,
PROB1^OUTCOMES.OUTCOMES.01,NULL,1,ISSELECTED);
ISSELECTED:=0;
IF $EXIST(PROB1"OUTCOMES.OUTCOMES.02) THEN ISSELECTED:=MCSELECTED;
MCITEM(P10UTCOMES,2,2,O,"RESPIRATORY RATE WNL FOR PATIENT",0,
PROB1"OUTCOMES.OUTCOMES.02,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1"OUTCOMES.OUTCOMES.03) THEN ISSELECTED:=MCSELECTED;
MCITEM(P10UTCOMES,3,3,O,"SPONTANEOUS VENTILATION",0,
PROB1"OUTCOMES.OUTCOMES.03,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1"OUTCOMES.OUTCOMES.04) THEN ISSELECTED:=MCSELECTED;
MCITEM(P10UTCOMES,4,4,O,"CHEST CLEAR PER AUSCULTATION",0,
PROB1"OUTCOMES.OUTCOMES.04,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1"OUTCOMES.OUTCOMES.05) THEN ISSELECTED:=MCSELECTED;
MCITEM(P10UTCOMES,5,5,O,"CHEST EXPANSION SYMMETRICAL",0,
PROB1"OUTCOMES.OUTCOMES.05,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1"OUTCOMES.OUTCOMES.06) THEN ISSELECTED:=MCSELECTED;
MCITEM(P10UTCOMES,6,6,O,"ABGS MNL FOR PATIENT ON ROOM AIR OR SUPP O2",0,
PROB1"OUTCOMES.OUTCOMES.06,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1"OUTCOMES.OUTCOMES.07) THEN ISSELECTED:=MCSELECTED;
MCITEM(P10UTCOMES,7,7,O,"INDEPENDENTLY HANDLES SELF-CARE ROUTINES",0,
PROB1"OUTCOMES.OUTCOMES.07,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1"OUTCOMES.OUTCOMES.08) THEN ISSELECTED:=MCSELECTED;
MCITEM(P10UTCOMES,8,8,O,"PERFORMS NECESSARY SELF-CARE ROUTINES",0,
PROB1"OUTCOMES.OUTCOMES.08,NULL,1,ISSELECTED);

ISSELECTED:=0;
IF $EXIST(PROB1"OUTCOMES.OUTCOMES.09) THEN ISSELECTED:=MCSELECTED;
MCITEM(P10UTCOMES,9,9,O,"COMMENT:(FREE TEXT ENTRY)",0,
PROB1"OUTCOMES.OUTCOMES.09,NULL,1,ISSELECTED);

FKEY:=KEYSET(ESCAPE,CSF4,CSF10);
DROSS0:=FMCREAD(FKEY,O,P10UTCOMES,NULL,NUM"ANS");
IF DROSS0<>13 THEN RETURN DROSS0;
MCHIDE;

IF NOT $VAL(SELECT[1,1])
THEN $EXIST(PROB1"OUTCOMES.OUTCOMES.01):=FALSE;
IF NOT $VAL(SELECT[1,2])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.02):=FALSE;
IF NOT $VAL(SELECT[1,3])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.03):=FALSE;
IF NOT $VAL(SELECT[1,4])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.04):=FALSE;
IF NOT $VAL(SELECT[1,5])
THEN $EXIST(PROB1^OUTCOMES.OUTCOMES.05):=FALSE;
MOVE*CODES BUFFER PROB1^OUTCOMES.OUTCOMES;

IF $VAL(SELECT[1,6]) THEN MOVE*CODES BUFFER PROB1^OUTCOMES.06;
IF $VAL(SELECT[1,7]) THEN MOVE*CODES BUFFER PROB1^OUTCOMES.07;
IF $VAL(SELECT[1,8]) THEN MOVE*CODES BUFFER PROB1^OUTCOMES.08;
IF $VAL(SELECT[1,9]) THEN MOVE*CODES BUFFER PROB1^OUTCOMES.09;
END;
REFERENCES


10. HUFF, S. M., CRAIG, R. B., GOULD, B.L. CASTAGNO, D.L. and SMILAN, R.E. A medical data dictionary for decision support applications. AT&T - Proprietary.


