Providing Reusability and Learning Support in the Simulation Model Development Environment

Jay D. Beams and Osman Balci

TR 92-03

Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

March 1, 1992

ABSTRACT

The Premodels Manager, one of the tools of the Simulation Model Development Environment (SMDE), is required to enable a user to locate and reuse components of successfully completed simulation studies and learn from past experience. This paper presents the SMDE Premodels Manager and describes how it provides reusability and learning support. Objectives are set forth and a design is established and implemented on a Sun workstation. The SMDE Premodels Manager consists of four working windows, three access windows, and three support windows. It uses the SMDE Premodels Database which is a highly modifiable repository of documentation on successfully completed simulation studies. It is evaluated with respect to the design objectives and is shown to provide effective reusability and learning support within the SMDE.

CR Categories and Subject Descriptors: I.6.7 [**Simulation and Modeling**]: Simulation Support Systems—*Environments*; I.6.8 [**Simulation and Modeling**]: Types of Simulation—*Discrete event*

Additional Key Words and Phrases: Database management systems, human computer interaction, learning support, reusability, simulation environments

TABLE OF CONTENTS

1. INTRODUCTION

Year after year considerable time and effort are spent conducting simulation studies without reusing components of previous studies or learning from the earlier experiences. Often a branch of a large organization (e.g., Department of Defense) starts a simulation study from scratch without benefiting from an earlier study conducted by another branch of the same organization. Millions of dollars can be saved by providing a repository of documentation on successfully completed simulation studies for the purpose of reuse and learning support.

The Premodels Manager (PM) software tool has been developed as part of the Simulation Model Development Environment (SMDE) to provide such reuse and learning support. The SMDE research project has been ongoing since 1983 under the objective of providing an integrated and comprehensive collection of computer-based tools to: (1) offer cost-effective, integrated and automated support of model development throughout its entire life cycle; (2) improve the model quality by effectively assisting in the quality assurance of the model; (3) significantly increase the efficiency and productivity of the project team; and (4) substantially decrease the model development time [Balci 1986; Balci and Nance 1987a,b; Frankel and Balci 1989; Balci et al. 1990; Bishop and Balci 1990]. The SMDE can also be characterized as a Simulation Support Environment or a Computer-Aided Simulation Software Engineering Environment. (The reader is recommended to read [Balci and Nance 1987a] to obtain the background information on the SMDE which is not repeated here in order not to prolong the length of this paper.)

Within a domain-independent discrete-event SMDE, locating and reusing information associated with previous simulation studies is very important, but also very difficult. Reusing components of and information about successfully completed simulation studies is important because it saves time and effort, and thus, reduces the overall cost of a simulation study. Even if the located information cannot be reused, a simulation analyst can learn from the past experience.

The difficulty associated with locating information lies in both the diversity of the information being managed and the diversity of the SMDE users. The problem of a user locating information which is available and which is actively sought sounds simple, but it is not. An example quickly illustrates some of the problems associated with locating reusable information. Consider locating information within a phone book. Although last names, first names, addresses, and phone numbers are all available, locating specific information is nearly impossible if a person's last name is not known. This trivial example, while adequate to illustrate one type of problem associated with locating and reusing information, only touches on the difficulties associated with reusing information within a more complex environment.

The purpose of this paper is to present the Premodels Manager software tool and describe how it is designed, implemented, and evaluated, and how it provides reusability and learning support. Section 2 describes the design objectives, hardware and software environment, Premodels Database, and components of the Premodels Manager. The evaluation is presented in Section 3. In Section 4, concluding remarks are given and future research is described.

2. PREMODELS MANAGER

The Premodels Manager (PM) is a software tool within layer 2 of the SMDE that works with the Premodels Database within layer 1 of the SMDE. This section describes the design objectives, hardware and software environment, Premodels Database, and components of the PM.

2.1 Design Objectives

The overall goal of the PM is to enable the user to: (a) locate and reuse components of successfully completed simulation studies, and (b) learn from past experience. The objectives identified below are intended to meet this overall goal:

❶ Provide easy methods of installing and maintaining documentation of successfully completed simulation studies in the Premodels Database.

❷ Provide appropriate methods of access to documentation of successfully completed simulation studies in the Premodels Database. Initiative, mechanisms, and complexity of access should vary according to task and type of user.

❸ Provide a stratified display, capabilities for copying and pasting, capability for storing in a user-created file, and printing of the information located by a user in the Premodels Database.

❹ Provide a user interface that satisfies the nine usability principles for interfaces [Nielsen 1990]: simple and natural dialogue, speak the user's language, minimize the user's memory load, consistency, provide feedback, provide clearly marked exits, provide shortcuts, good error messages, and prevent errors.

❺ Provide context-sensitive help that is always available in a consistent manner. The system should use all available information on the user's state and avoid placing the burden on the user.

2.2 Hardware and Software Environment

The PM is implemented on a Sun color computer workstation running the Sun UNIX operating system. The Sun workstation consists of a 380-MB disk subsystem, a 1/4-inch cartridge tape subsystem, 8-MB of main memory, a 19-inch color monitor with a resolution of 1152 x 900 pixels, a pointing device called a mouse, an Apple LaserWriter laser printer, and a connection to the Ethernet network which enables high speed file transfer to and from other University computing systems.

The most significant element of the user interface is the 19-inch bit-mapped display screen. This display technology offers excellent graphics capabilities. The user communicates with the Sun via keyboard input and the mouse. The mouse may be used to point to or select locations on the screen. The system provides feedback on the current mouse location by continuously updating the position of the mouse pointer. Virtually any material that is displayed on the screen can be pointed to and treated as input.

The windows are the basic building block of the user interface and are roughly analogous to sheets of paper on a desktop. Windows may be resized, overlapped, closed, or quit. Resizing a window allows the user to effectively increase the available workspace. Overlapping is useful when the user needs to interact with multiple windows, but there is not enough room to display all of them. Closing reduces a window to a small 64 x 64 bit image known as an "icon". Icons are useful when a window contains a process that does not require immediate user interaction, but it is still desirable to keep that process running (shell, for example). Quitting a window destroys the window entirely when there is no further use for it.

Menus and buttons are perhaps the most important window features in terms of the user interface. Menus provide a user with the ability to select an option from a finite list of choices. Menus may be "pop-up", requiring the user to depress a mouse button before they are active. Buttons allow the user to interact in a more limited way with the running program. Selecting a button requires the user to position the mouse cursor over the button image and depress a mouse button. This action executes a C function that has been associated with the button by the programmer.

The PM software consists of 19,876 lines of documented SunView, C, and EQUEL/C. All window features are programmed using the SunView graphical user interface software which consists of high-level object-oriented routines [Sun Microsystems 1988]. All storage and retrieval of data are achieved by the use of INGRES relational database management system [Sun Microsystems 1986]. The use of INGRES is completely hidden from the PM user. EQUEL/C (Embedded QUEry Language)/C is used to access INGRES within C and SunView programs. Although many modern computer systems have the hardware capability to support the PM, the SunView interface technology has proved invaluable for its rapid development.

2.3. Premodels Database

The Premodels Database (PD) contains documentation on successfully completed simulation studies. The PD Administrator is responsible for storing the documentation of a simulation study in the database with respect to its highly modularized Table of Contents using the Installer component of the PM. The Table of Contents is not fixed and is prepared individually for each simulation

study.

The PD has two types of data: (a) data stored in records of PD tables (e.g., completion date of a simulation study), and (b) data stored in text files outside of the PD (documentation text).

The PD is composed of *tables*. The columns of a table represent information categories and are called *fields*. The rows in a table are called *records*. Records are reusable information being managed by the PM. An individual cell in a record, which is defined by a field in the table, is called a field value, or simply a field if context clearly indicates whether "field" refers to a column or cell. Tables and fields have classifications associated with them that determine how they are to be managed by the PM.

Two types of tables exist in the PD: user tables and system tables. User tables contain the actual information associated with simulation studies and can be easily modified without changes to the PM. System tables save qualifications which allow only certain records from user tables to be displayed and, unlike the user tables, are required for the correct operation of the PM. Although the system tables are not directly accessible by general PM users, information may be manipulated in these tables by the system administrator.

Although the PD is built on a database management software (INGRES) that provides low-level types such as two byte integers, from a design standpoint, as well as from a user's perspective, fields in the PD have high-level field types associated with them such as Key, Link, Text-File, or Executable-File.

The first field in every table is a special field type called *Key*. Each record in a table has a unique value for its Key field, thus each record in the PD can be identified by a user and by the PM. Operations on records and relationships between records are all based on the value of the Key field of the record. Because Key fields uniquely identify each record in the PD, another special field type called *Link*, can be used to link one record to another. A Link field in a table is statically associated with another table. The value of a Link field is simply the Key field value of a record in the linked table. Since Link fields are not necessarily unique, a one-to-many relationship is defined from the Key field of one record to the Link fields of other records, and conversely, a many-to-one relationship exists from the Link fields of records to the Key field of another record. By combining many-to-one and one-to-many relationships, numerous other relationships can also be specified.

Text-File and Executable-File are two other important field types. Text-File fields contain the name of a text file and the position in the text file that should be displayed. Although the actual text file exists outside of the PD, conceptually, it is simply part of the Text-File field value. An Executable-File field contains either an executable operating system command or the name of a file which is executable. The Executable-File field can be used for future multimedia extensions such as

playing recorded sounds and showing a video of the animation of a simulation model.

The PD is not an inflexible storage structure and is not tightly intertwined in the design or implementation of the PM. Instead the PD is a highly modifiable reservoir of information whose current configuration is irrelevant to the PM with the exception that its structure must conform to the standards described above.

2.4. Components of the Premodels Manager

The PM consists of a collection of windows which work together to allow different types of interactions between users and the PD. Three types of windows, shown in Figure 1, are used in the PM. The windows with the thick borders are the working windows. The windows with the double-lined border are the access windows which guide a user to an appropriate working window. The other windows with a single-lined border are support windows which assist the working windows in performing specific tasks. Not shown in Figure 1 is the Help support window which is accessible from any access or working window.

All users enter and exit the PM through the Driver access window from which Retriever and Administrator access windows can be activated. The Retriever provides access to either the Browser or Searcher working window. The Administrator provides access to either the Installer or Maintainer working window. Unlike the access and working windows, the support windows are displayed simultaneously with other windows.

2.4.1. Working Windows

Four working windows currently exist in the PM: *Browser*, *Searcher*, *Installer*, and *Maintainer*. Each working window in the PM minimally contains a control panel, which provides icons and items to control the window, and a display panel, which actually displays records from the PD. A working window control panel contains the standard selectable "Return" icon to return to the calling window, the standard selectable "Help" icon to toggle the display of help, and other icons and items associated with the working window. A working window display panel displays records, and relationships between records, in a manner appropriate to the specific working window. The fields associated with a record are presented as a vertical list within the working window display panel. If the record is modifiable by a user, as with the Installer working window, then the field values are directly editable by a user. If the record is not modifiable, as in the Browser and Searcher display panels, then field values are neatly formatted to be read by a user, but can not be edited.

Relationships among records in a display panel are shown both through Link and Key field values and through the indentation and ordering of records in the display panel. Two menus are

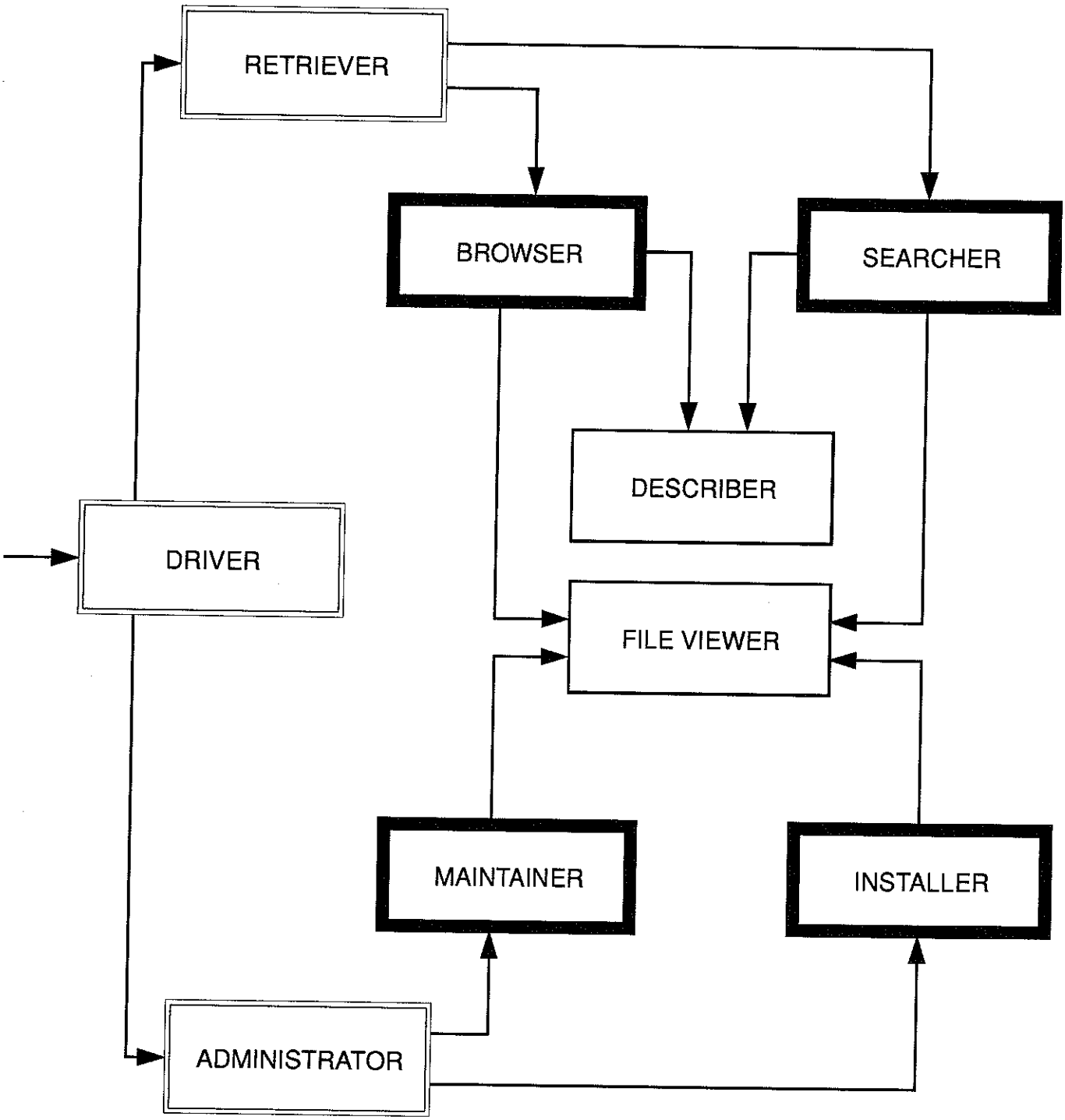

Figure 1. Premodels Manager Window Access Hierarchy

associated with each field and field value of a record. One menu contains options associated with a specific field in the record and the other menu contains options associated with the entire record. This consistent approach to the displaying of records within all working window display panels allows the experience gained by using one working window to be directly applicable to other working windows.

2.4.1.1. Browser Working Window

Browser provides both a method of browsing all records within a table and access to the QUEL query language to perform queries on a table. Browser, shown in Figure 2, is divided into four subwindows: Browser Control Panel, Browser Display Panel, Browser Range Panel, and Browser Qualify Panel.

Browser Control Panel contains two items and four icons to control the other Browser subwindows. The first item, called the "Table Name" item, displays the name of the current table being browsed and the second displays a description of this table. The "Table Name" item has a menu attached to it which allows a user to select a new table as the current table. It may also be selected to refresh Browser Display Panel, Browser Qualify Panel, and Browser Range Panel. The four icons in Browser Control Panel include the standard selectable "Help" and "Return" icons along with two other icons labeled "Do Qual" and "File Qual." The "Do Qual" icon, when selected, performs the qualifications indicated in Browser Range Panel and Browser Qualify Panel. The "File Qual" icon may either be selected to pop up Filing Control Panel, or may have its menu accessed which provides quick access to the options available in Filing Control Panel. Filing Control Panel provides an item, called "User Name", in which a user may type in the name of a Browser configuration, and four icons called "Load Qual", "Save Qual", "Delete Qual", and "Abort". The "Load Qual" icon, when selected, loads the Browser configuration indicated by the "User Name" item. The "Save Qual" icon, when selected, saves the current Browser Range Panel and Browser Qualify Panel to the Browser configuration indicated by the "User Name" item. The "Delete Qual" icon, when selected, deletes the Browser Configuration indicated by the "User Name" item. The "Abort" icon, when selected, simply removes Filing Control Panel from the screen without performing any changes.

Browser Display Panel provides a simple, yet powerful, method of displaying records and relationships between records retrieved from the PD. Records are always displayed in their entirety, i.e., all fields of a record are displayed. The fields in a record are listed vertically and are presented with both the name of the field and the field value. Field names are presented using the notation "table.field" where table is the name of a table in the PD. Field values are presented based on the type of field being displayed. Relationships between records are displayed both by the values asso-

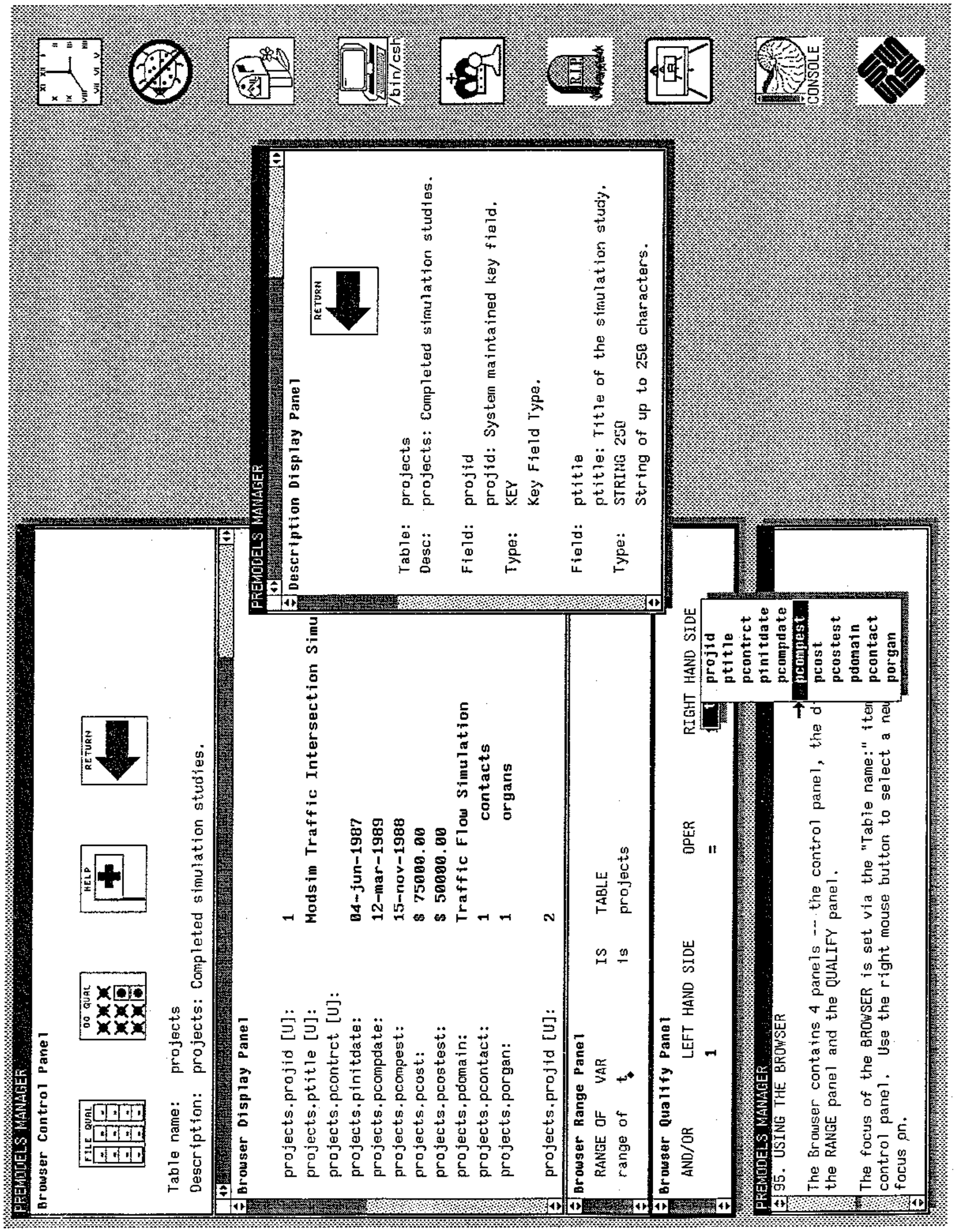

Figure 2. Browser Working Window and Describer Support Window

ciated with Key and Link fields and, more visually, through indentation. Two types of general relationships exist between tables called one-to-many (1-M) and many-to-one (M-1). An M-1 relationship means that a Link field in the current record is associated with the key field in another record. A 1-M relationship means that the Key field of the current record is associated with the Link field in other records. A special type of 1-M relationship handles relationships, such as a table of contents, in which only linked records at a top level should be initially presented. 1-M and M-1 relationships, as well as other options, are available through menus associated with records in Browser Display Panel.

Browser Range Panel and Browser Qualify Panel provide a simple interface to the QUEL query language. *Browser Range Panel* allows range variables to be mapped to tables in the PD. To be consistent with QUEL, a range variable is presented in Browser Range Panel as "range of t is table" where "t" is the name of some range variable and "table" is the table in the PD. Only the name of a range variable may be directly modified by a user. To change the name of the table, a new range variable must be added associated with the desired table. Each item in Browser Range Panel provides a menu consisting of two options called "Append" and "Delete". The "Append" option contains a submenu which has the names of all available tables with which range variables may be associated. The "Delete" option simple deletes the range variable from which it was accessed. Range variables specified in Browser Range Panel may then be used in Browser Qualify Panel to form potentially very complex queries.

Browser Qualify Panel is used to represent a qualification on records which will be retrieved from a table. A qualification is divided into individual conditions which are combined together using "and" and "or" operations. Each condition contains three parts: a left hand side, a right hand side, and a comparison operator. Browser Qualify Panel allows each component in a qualification to either be typed in by a user, or selected from a menu. The menus associated with the left hand side and right hand side allow fields associated with range variables to be easily entered; however, to enter a value in the left hand side or right hand side that is not a field reference, a user must type in the value. Each item in Browser Qualify Panel also has a second menu associated with it that provides the options "Append", "Delete", and "Reset". The "Append" option is used to add conditions to a qualification. The added condition initially has the form "and 1= 1" which does not affect the value of the entire qualification. The "Delete" option removes a condition, from which it was accessed, from a qualification. The "Reset" option resets all the values in a condition to their values when the condition was initially displayed or when the qualification was last saved, whichever is more recent. The qualification in Browser Qualify Panel is performed when the "Do Qual" icon in Browser Control Panel is selected, at which point, Browser Display Panel is appropriately updated.

2.4.1.2. Searcher Working Window

Searcher, shown in Figure 3, displays records located within the PD like Browser; however, unlike Browser, Searcher locates records relative to a value without regard to the table in which records reside. Searcher contains two subwindows called Searcher Control Panel and Searcher Display Panel. As with each working window, the display panel displays records associated with the PD and the control panel indicates which records to display in the display panel.

Searcher Control Panel contains three items and four icons used to identify desired records. The three items are: (a) a text item in which a user can type a character string to be looked for in records, (b) an item which indicates whether or not files should be searched, and (c) an item which indicates whether or not the search is case sensitive.

In addition to "Return" and "Help" icons, a selectable "Clear Search" icon is provided to clear the search indicators in Searcher Control Panel and the records in Searcher Display Panel, and a selectable "Do Search" icon is provided to perform the search indicated by the three items.

Information in Searcher Display Panel is displayed in a manner identical to Browser Display Panel. In this way, although Browser and Searcher provide very different ways of locating records, a user's experience with one window is highly reusable for the other window. Both Searcher and Browser display records that cannot be modified by the user, thereby protecting the PD from unauthorized updates; however, in order to allow records to be added or updated, another working window, called Installer, is provided.

2.4.1.3. Installer Working Window

Although records are displayed somewhat differently in Installer to allow for updating record values, the general manipulation of records in Installer is very similar to the other working windows. Installer, shown in Figure 4, contains two subwindows called Installer Control Panel and Installer Work Panel.

Installer Control Panel allows a user to save or remove the records displayed in Installer Work Panel. Installer Work Panel displays records which can be modified by a user. Most information can be installed in the PD using Installer; however, files referenced by records must be edited and copied to an appropriate directory outside of the PM. Installer Control Panel contains a selectable table name item and an associated table description item which function as in the Browser window to set the initial table from which to display records. The work mode item indicates whether the records in the indicated table are browsed or if a particular record, such as a new record, is selected. The control panel contains four icons: "Return", "Help", a selectable "Delete Record" icon to delete the

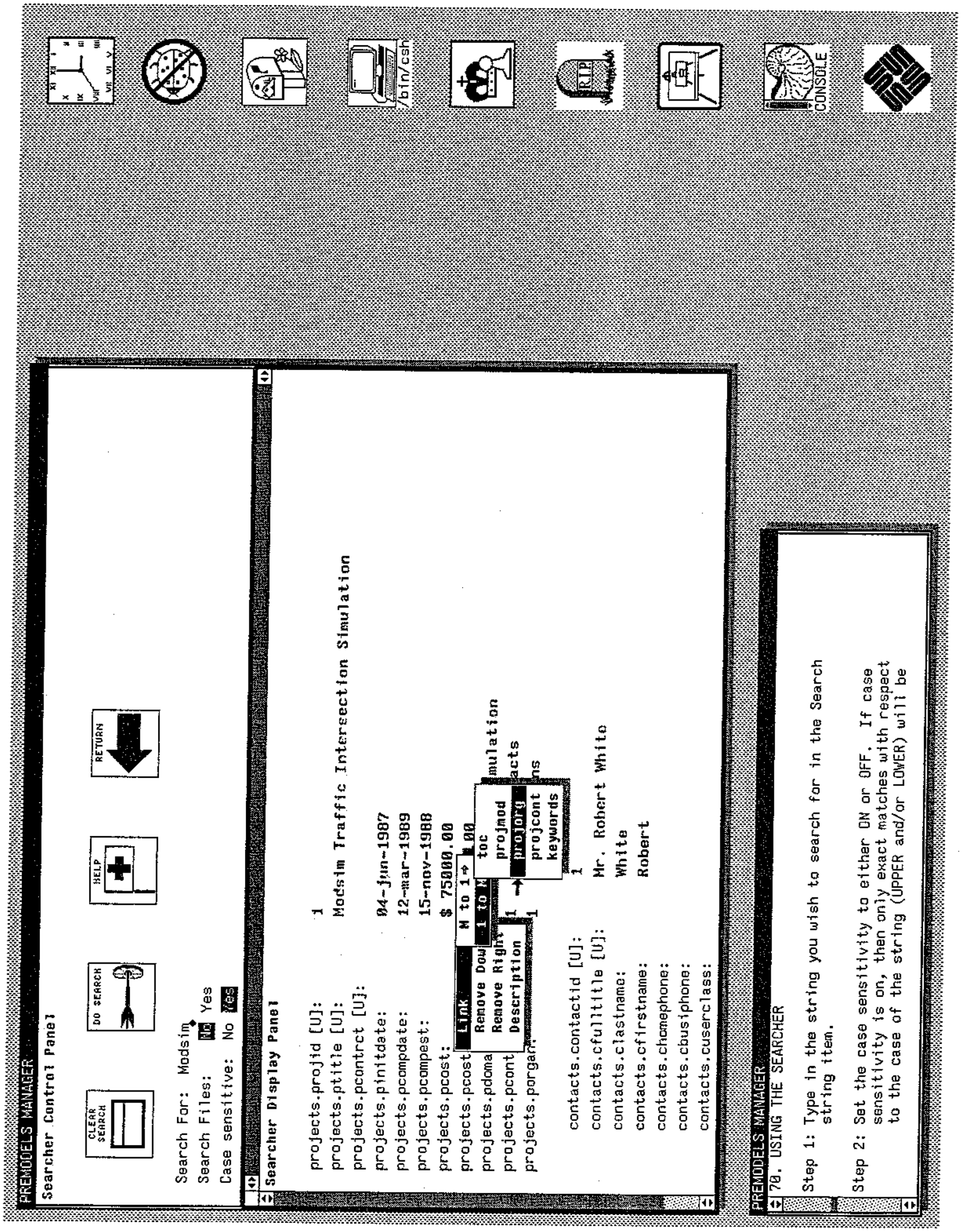

Figure 3. Searcher Working Window

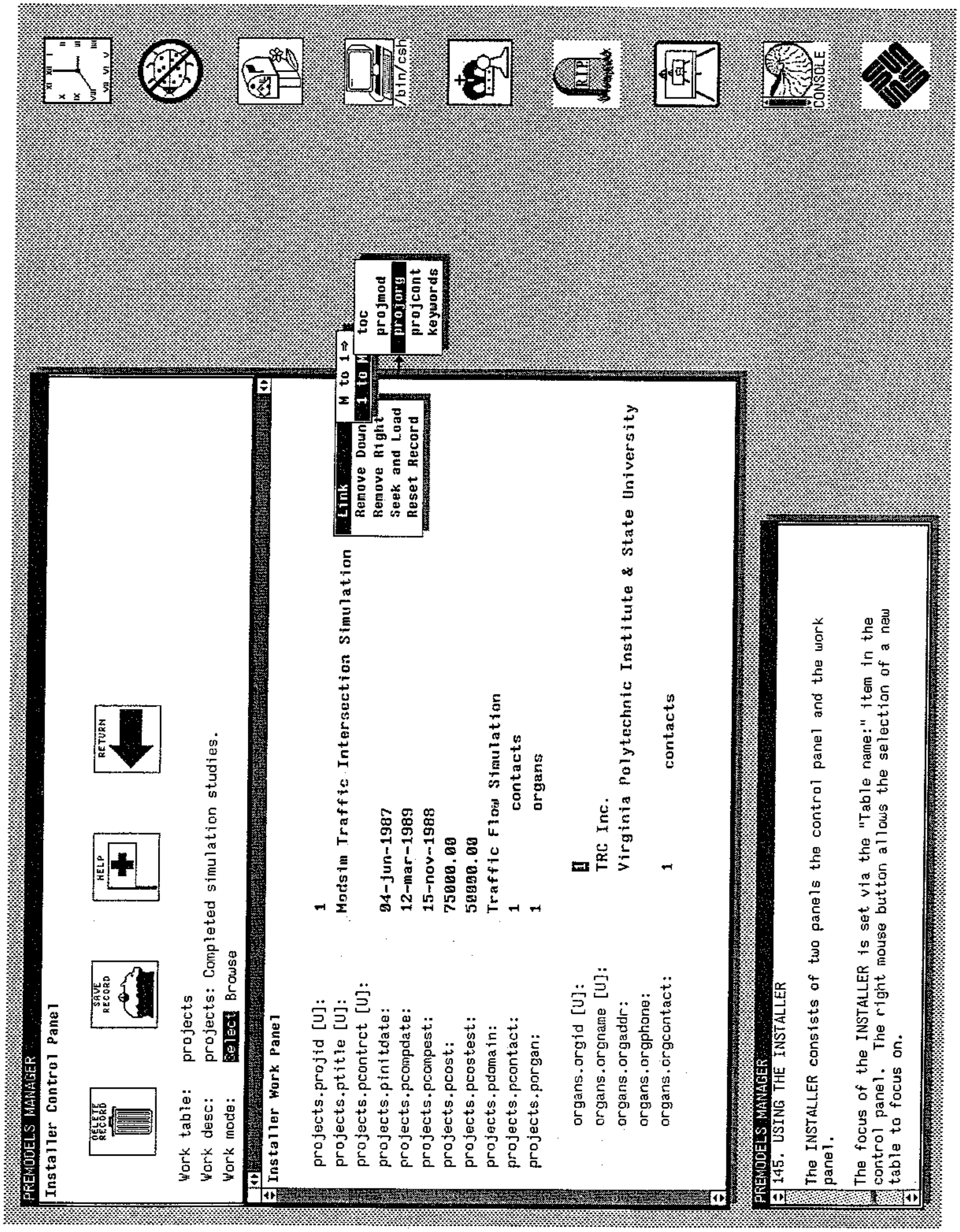

Figure 4. Installer Working Window and 1-M Link Menu

records currently displayed in Installer Work Panel, and the selectable "Save Record" icon to save the records displayed in Installer Work Panel.

Installer Work Panel, unlike Browser Display Panel and Searcher Display Panel, displays records which are editable. The first field in a record is a Key field and is used to uniquely identify records during "Save Record" and "Delete Record" actions. Even though records are editable by the Installer, the basic method of displaying the relationships between records are identical to the Searcher and Browser display panels.

2.4.1.4. Maintainer Working Window

Maintainer window enables the addition and modification of tables in the PD. Its control panel provides a method of specifying a table name similar to Installer and Browser except that in addition to selecting a table name from a menu, a table name may be typed in. Similarly, the description of a table may be modified by simply entering the new table description.

Table names inputted may or may not exist in the PD. If an inputted table name exists in the PD, such as the "projects" table displayed in Figure 5, then the table's description is displayed in the control panel and a list of fields comprising the table is displayed in the display panel. If an inputted table name does not exist in the PD, then the table's description is left blank and a single key field is displayed within the display panel. The name of this key field defaults to the table name plus the extension "id"; however, this key field name may be modified.

Fields may be added, deleted, or modified using menus associated with the "Table name" item and with the "Field" items (Figure 5). A field may be modified by entering a new field name, a new field description, or selecting a new type using the "Field" item menu. Changes in field names and field descriptions do not affect records currently stored in the table; however, changing a field's type removes any data currently stored in the field. Changes to a table are not recorded until the return icon is selected.

2.4.2. Access Windows

The access windows, shown in Figure 1 with a double lined border, enable a user to select a working window in a two-level hierarchy. The first level consists of Driver and the second level consists of Administrator and Retriever. Each access window consists of a single control panel with four icons and associated selectable messages including "Help" icon and an icon to exit the window or, in the case of Driver, to exit the PM.

Driver window, shown in Figure 6, is the highest level window of the PM. Retriever window, also shown in Figure 6, can be selected by clicking on the "Retriever" icon. Administrator window,

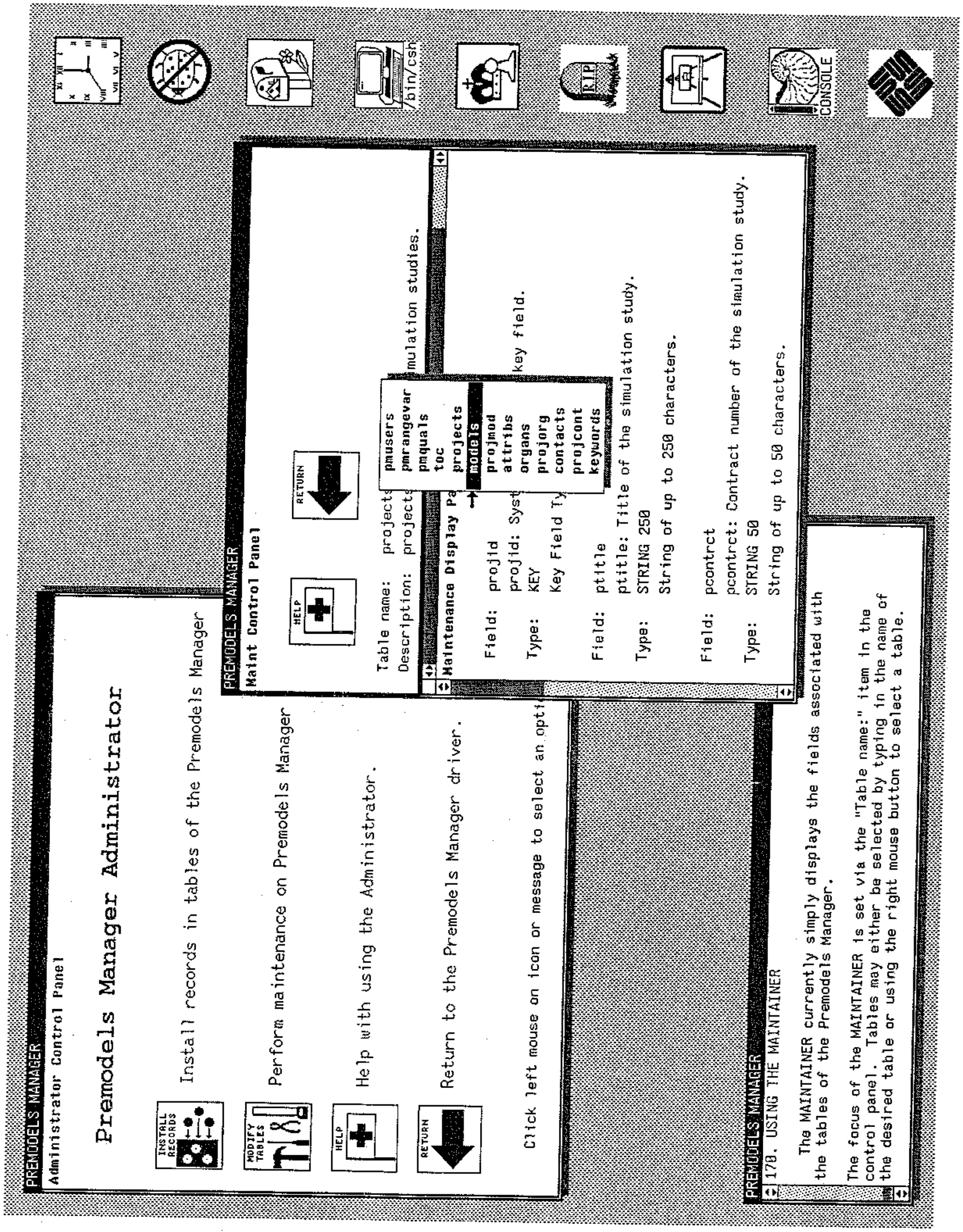

Figure 5. Administrator Access Window and Maintainer Working Window

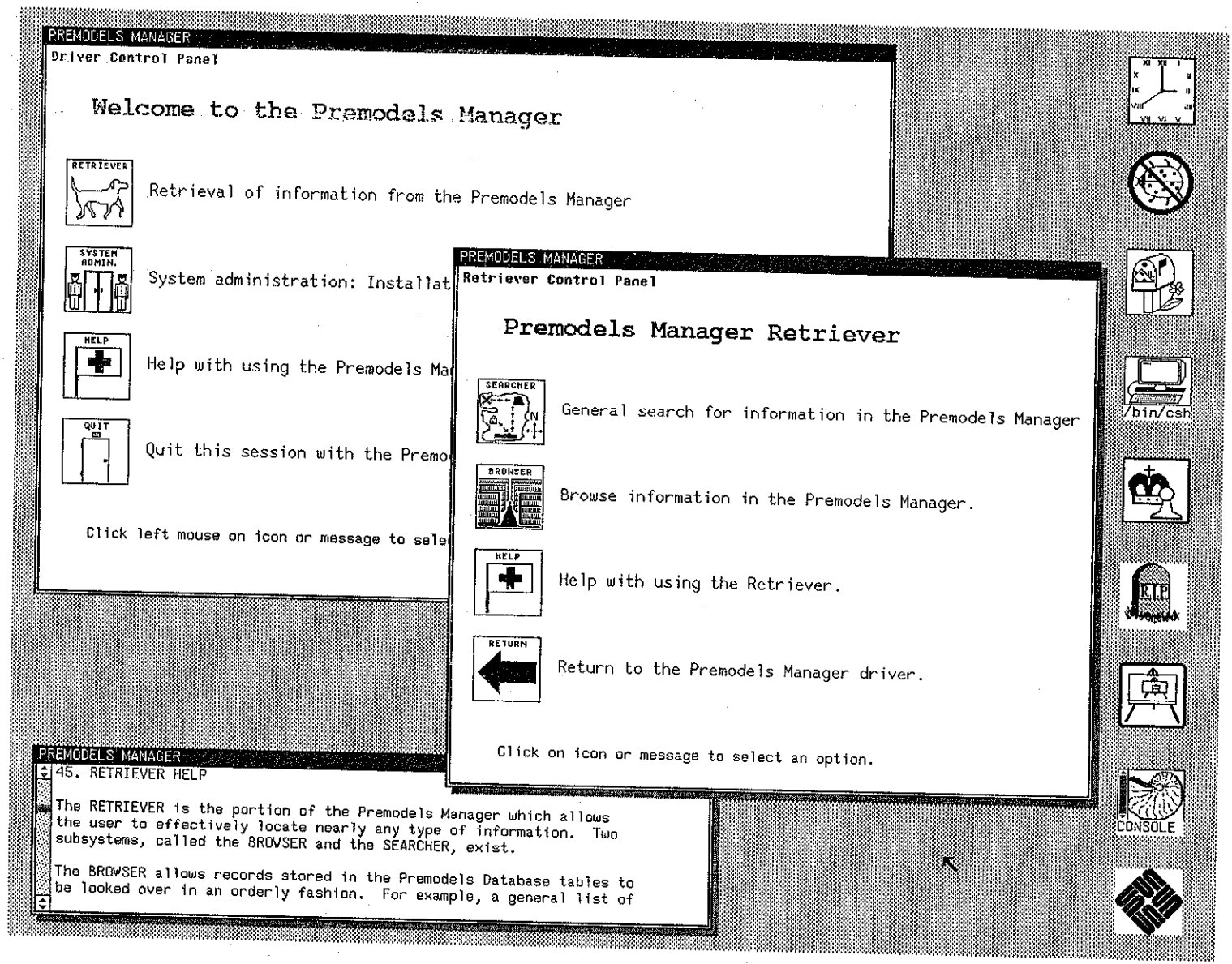

Figure 6. Driver and Retriever Access Windows

shown in Figure 5, can be selected with a special password window by clicking on the "System Administrator" icon. Retriever provides access to Browser and Searcher and is available to all users. Administrator provides access to Installer and Maintainer and is available to only those users who are authorized to add, delete, or edit records.

2.4.3. Support Windows

File Viewer, Describer, and Helper windows support the operations of the access and working windows.

File Viewer, shown in Figure 7, is accessible from all working windows: Browser, Searcher, Maintainer, and Installer. In each working window, File Viewer is accessible via a menu associated with text file fields in a displayed record. It consists of File Viewer Control Panel and a scrollable subwindow for displaying the file in read-only mode. File Viewer Control Panel facilitates determining a file's reusability and reusing the file if warranted. It has "Save File", "Print", and "Return" icons. "Save File" icon is used to save the displayed file in a user file. "Print" icon is used to print the displayed file. "Return" icon removes the File Viewer. The "Search" item is used to rapidly locate a specified character string within the displayed file.

Describer, smaller window in Figure 2, is accessible through the menu associated with a record displayed in Browser Display Panel or Searcher Display Panel. It describes a table with its associated fields. "Return" icon removes it. It can be updated either by selecting a different record from a working window or directly through the "Table name" item.

Helper, the lower left corner window in Figures 2–6, displays context sensitive help text when the "Help" icon is selected. Once displayed, the Help window continues to be displayed, even if windows are entered and exited, until the Help icon is again selected. Helper contains a single scrollable subwindow which displays specified positions in the read-only help file. Helper window is displayed and removed by clicking on "Help" icon. Rather than updating the position in the help file when the "Help" icon is selected, the position is updated automatically each time a working or access window is entered or exited. Thus, a user may display Helper from any window and have appropriate help text displayed as other windows are entered and exited.

3. EVALUATION OF THE PREMODELS MANAGER

The PM has been evaluated with respect to the five design objectives of Section 2.1 and has been found to provide effective reusability and learning support within the SMDE. The design objectives altogether contribute to enabling the user to locate and reuse components of successfully

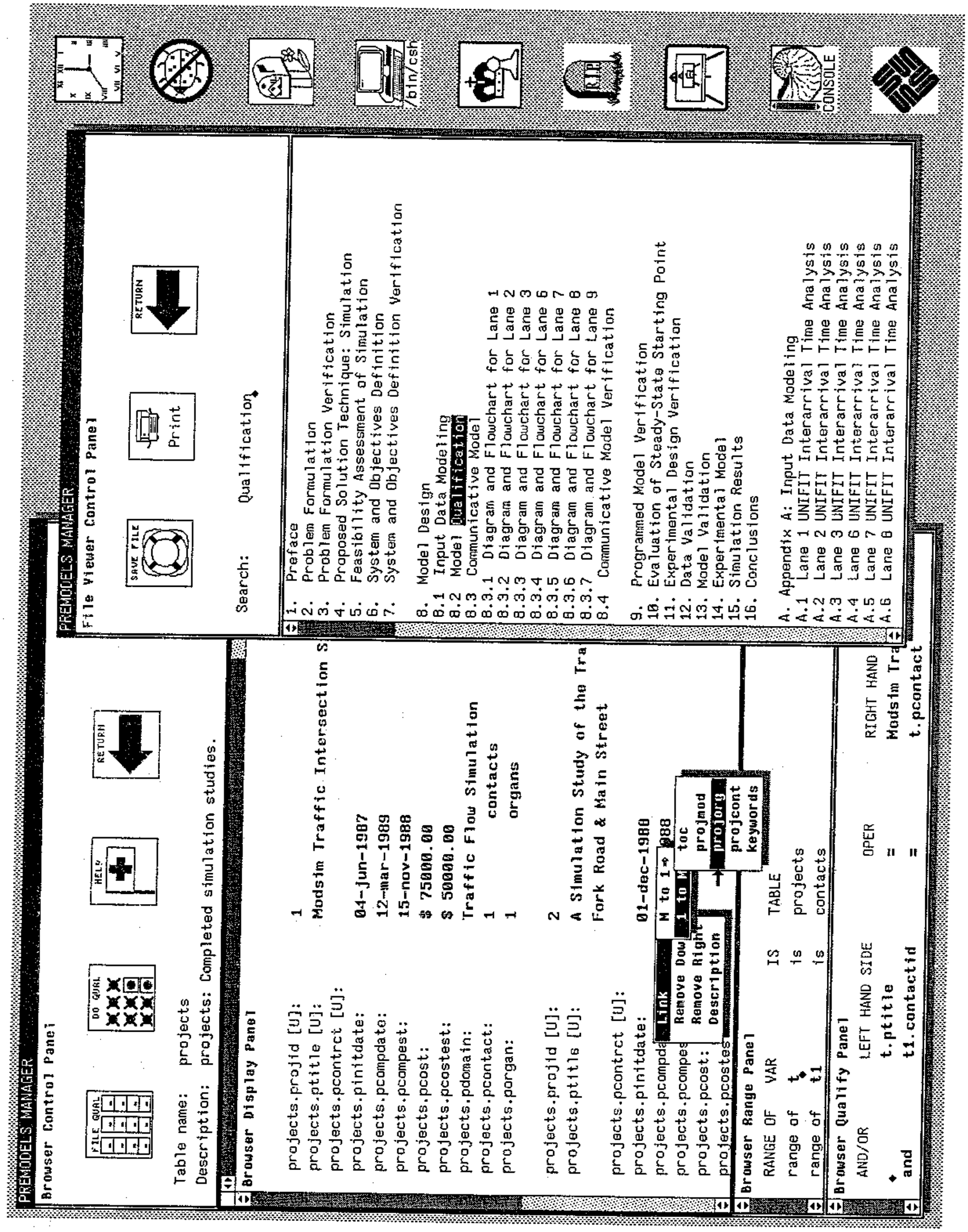

Figure 7. File Viewer Support Window

completed simulation studies and learn from past experience. The evaluation is reported below for each design objective.

3.1. Objective 1

> "Provide easy methods of installing and maintaining documentation of successfully completed simulation studies in the PD."

Objective 1 is evaluated from two perspectives: evaluating the installation of documentation of a simulation study and evaluating periodic maintenance of data stored in the PD.

Installing a simulation study documentation in the PD can be an arduous task. First, the files containing the documentation must be copied to appropriate locations. Second, records must be filled with data extracted from the documentation. Third, internal and external relationships among records must be defined. The PM does not assist in the first task. It assists in the second and third tasks by: (1) using the table of contents of the documentation to help guide installation, (2) allowing files to be viewed and file locations to be specified during the installation, (3) reflecting internal relationships among records in a stratified display, and (4) predefining relationships among records based on tables and fields so that external relationships need not be explicitly enumerated. The major problem encountered during the installation is typing in each line of the table of contents. An automated method of reading lines of a table of contents directly from the simulation study documentation would simplify the installation.

Maintenance of records in the PD can easily and quickly be accomplished. This is made possible by several design decisions: (a) The same screen used for installation is also used for maintenance and thus all capabilities available during installation are available during maintenance, (b) The PM uses a single record save operation instead of record update and record append operations, (c) The PM provides both a select mode for locating a single record for modification and a browse mode for updating a collection of records, and (d) The PM can restore the initial values of a modified record without aborting changes to other records.

3.2. Objective 2

> "Provide appropriate methods of access to documentation of successfully completed simulation studies in the PD. Initiative, mechanisms, and complexity of access should vary according to task and type of user."

The PM provides various methods for accessing the documentation. For accessing simple facts, the PM locates appropriate PD records based on the relationships among records. A stratified display assists in using these relationships by allowing each relationship to be visualized. The PM

provides three basic methods of locating the root records in a stratified display. These methods are designed to serve different user needs. First, if a related value is known, a user can simply search the PD for a character string and display all records containing the string. This method is appropriate for users who are unfamiliar with the PD structure. Second, if the type of information is known, a user can browse the records in a table. This method assumes a user to have some concept of the PD tables; however, even a first time user can rapidly browse numerous tables to see if any of the records are appropriate. Third, a user can use a QUEL-like query language to query a table for appropriate records. This method requires a user to have some knowledge of a database query language.

For more complex information, the PM does not only locate records, but also uses the records to access text files which may in turn be searched by using the File Viewer. A stratified display may be used to locate and display a table of contents and then the appropriate table of contents records may be used to access locations in a text file. Alternatively, a character string may be searched for in all files in the PD. If the string is located, then a record associated with the file is displayed and the file may subsequently be accessed. Once a file is displayed, it may be individually searched for a character string.

The major problem with the file search mechanisms in the PM is that the search is character string-based rather than concept-based. In other words, if a file is searched for "car" it will not find "automobile." This is a major limitation given the diversity of terminology used within simulation studies.

3.3. Objective 3

> "Provide a stratified display, capabilities for copying and pasting, capability for storing in a user-created file, and printing of the information located by a user in the PD."

In order to display records located in the PD and the relationships among records, the PM uses a stratified display. Although relationships are maintained using key and link fields, the stratified display allows a user to quickly visualize relationships. Any number of nested relationships can be displayed; however, a specific record can only have one nested relationship displayed at one time. For example, a record containing a simulation study title is linked to both organizations and people; however, only its organizations or its people may be displayed at one time.

Text files containing the documentation of successfully completed simulation studies are displayed using the File Viewer. Since documentation in the PD is archival, it may not be modified by PM users; however, it may be copied to a user file and then manipulated using standard text editing facilities. A located text file may also be printed. Copying and pasting parts of a file is possible

through the use of the SunView graphical user interface.

3.4. Objective 4

> "Provide a user interface that satisfies the nine usability principles for interfaces [Nielsen 1990]: simple and natural dialogue, speak the user's language, minimize the user's memory load, consistency, provide feedback, provide clearly marked exits, provide shortcuts, good error messages, and prevent errors."

This section contains an evaluation of the PM's user interface with respect to each of the nine usability principles developed by Nielsen [1990].

Principle 1: Simple and Natural Dialogue

To be simple, dialogues "... should not contain information that is irrelevant or rarely needed" [Nielsen 1990]. Since the PM users (client, organization, project, and software development managers; analysts; and programmers) have diverse backgrounds and expertise, information which is irrelevant to some users is relevant to others. Since all information is presented as complete records, unless all fields in a displayed record are pertinent to a user, some irrelevant information is inevitably displayed. Although the PM cannot limit the fields displayed, it provides several methods for preventing irrelevant records and control items and icons from being displayed. One way in which the PM limits the amount of information presented to a user is by dividing operations between different working, access, and support windows rather than a single window with lots of options. The records presented in a display window is also limited by using a hierarchical approach. This approach initially displays top level records and, only on a user's request, are these top level records linked to other records. Two menu options associated with records, called "Remove Down" and "Remove Right", allow unnecessary records to be explicitly removed from a display by a user.

Principle 2: Speak the User's Language

By mostly using icons, rather than words, the PM is able to avoid many communication problems associated with users who have diverse backgrounds. When it is necessary to communicate with words, the PM uses common language and is careful to avoid domain specific terminology. Explanations in the form of messages and context sensitive help also attempt to overcome language barriers. The PM's language is simple; however, the importance of the PD mandates the use of several database terms such as table, field, and record. To simplify working with the tables and fields, the PM provides their descriptions.

Principle 3: Minimize the User's Memory Load

The PM user interface requires very little memorization. This is accomplished by using icons and menus for nearly all operations as well as providing consistency across all PM windows. The experience gained working with one PM window is directly applicable to other PM windows. On first access to the PM, a user is guided by very simple access windows to an appropriate working window. Each access window provides four icons. Each icon is accompanied by an appropriate explanation to remind a user what the icon does. Even if a user has forgotten how to select icons, a message at the bottom of the access window tells a user to click on an icon or message. The message does not specifically indicate that the left mouse button should be clicked; however, there is no harm in a user trying to click other mouse buttons before using the left one.

If a user selects the "System Administrator" option from Driver, a password must be entered to access Administrator. For obvious reasons, no help is given with entering the password other than a message indicating that a password is expected.

Once in a working window, control icons are presented in a control panel. Like the access windows, these icons are selected by clicking the left mouse button on them. Each icon is intended to remind a user of the icon's purpose; however, for clarity, each icon also contains a simple textual explanation.

A final feature that virtually eliminates the need to memorize any portion of the system is a special support window which provides context sensitive help. This window, called Helper, is available for every access and working window. A special "Help" icon is used to toggle the display of Helper. Once displayed, Helper is automatically updated each time a working or access window is entered or exited.

Principle 4: Consistency

One of the greatest strengths of the PM user interface is its consistency. Consistency is seen not only within an individual window, but also across all windows in the PM. The principle of consistency is incorporated into the design of the PM user interface in various ways. Some of the consistent elements are described below.

Every icon displayed may be accessed by clicking the left mouse button on the icon. Every access window contains icons, and messages are displayed within a single subwindow. Every working window contains a control panel at the top of the window which controls that window. Every access window and every working window contains the "Help" and "Return" icons. The "Help" icon always toggles the display of Helper. The "Return" icon always removes the current window and activates the window from which it was accessed. Every working window, except Maintainer,

contains a display panel in which records are displayed. Indentation is used in each of these windows to represent relationships between records. Browser and Searcher use identical display panels to work with records once they are located. All error messages are displayed using a single error message window. This window also uses the "Return" icon for its removal.

Principle 5: Provide Feedback

Feedback is generally provided by the PM immediately; however, certain complex queries in Browser, extremely general searches in Searcher, and the addition of a large number of records in Installer may be slow in providing feedback. Also, feedback is not provided on unsuccessful searches and queries.

Principle 6: Provide Clearly Marked Exits

The PM does not allow a user to directly exit the system from any window except Driver. Each window does however provide a clearly marked "Return" button. The decision to force all users to enter and exit through Driver was intended to protect users from inadvertently exiting the PM and having to wait for the system to reinitialize for another session.

Principle 7: Provide Shortcuts

In order to keep the design of the PM as simple as possible, very few shortcuts are provided. One shortcut which is available is a menu associated with the "File Qual" icon in Browser which allows the "Save Qual", "Load Qual", and "Delete Qual" functions to be accessed without going through Filing Control Panel. Similar shortcuts could be provided; however, "... the principle of providing shortcuts often conflicts with the principle of simple and natural dialogue" [Nielsen 1990]. The shortcuts are limited until more clearly defined user needs are identified through the extensive use of the PM.

Principle 8: Good Error Messages

Very few error messages are displayed by the PM because few errors can occur. Two error messages are: "unrecognized password" and "search string size is too short." These messages clearly indicate the problem and indirectly a solution. One problem with the PM is that it does not issue an error message on unsuccessful queries even if the user specified query is invalid. This could result in information not being located because a user incorrectly interprets a query as being unsuccessful rather than invalid.

Principle 9: Prevent Errors

Most errors are avoided in the PM due to the use of a menu and icon based interface rather than the traditional command line style interface. By using menus and icons, only appropriate options are available within each window and typographical errors are practically eliminated.

3.5. Objective 5

> "Provide context-sensitive help that is always available in a consistent manner. The system should use all available information on the user's state and avoid placing the burden on the user."

The context-sensitive help facility of the PM is simple, yet powerful. All access and working windows have a help icon. The support windows do not currently provide access to help; however, since support windows are always used together with working windows, help can be accessed through the associated working window. The help facility could be further refined by tracking not only the active window, but also other information about a user's state such as the last attempted operation, the current position within the active window, and the mode under which a window is currently operating.

4. CONCLUDING REMARKS AND FUTURE RESEARCH

The evolutionary prototyping software engineering approach has been used in developing the PM. Several PM prototypes have been developed, evaluated, and discarded prior to the current version described in this paper. Knowledge gained by experimenting with one prototype PM has been used in developing the next improved PM prototype.

The use of documentation of *successfully* completed simulation studies has been assumed in the research described herein; however, one can also learn from and reuse some components of *unsuccessful* simulation studies. The PM can be modified to display information with a label indicating whether the information is about a successful or an unsuccessful simulation study so that the user can interpret the information accordingly.

The integration of the PM with the other minimal SMDE tools, especially with the Model Generator, remains to be done. Currently, while using the Model Generator, the user can activate the PM as an independent process and use both by alternating from one to other. However, the Model Generator and the PM are not tightly integrated.

The most important future research should deal with multimedia and hypertext extensions. Documentation of a simulation study may include text, graphics, scanned images, sound, and video.

The PM should be able to: (a) display and print documents using a text formatting language such as SGML, LaTeX, and Troff; (b) display and print graphics and images in a variety of formats, e.g., TIFF, PICT, EPSF; (c) play recorded sounds; (d) show video clips, e.g., video of animation of a simulation model. With the availability of multimedia and hypertext features, the PM can provide a much more effective learning support and reusability than it is currently possible.

ACKNOWLEDGEMENTS

This research was sponsored in part by the U.S. Navy through the Systems Research Center at VPI&SU. The authors acknowledge stimulating discussions with R. E. Nance, J. L. Bishop, E. J. Derrick, E. H. Page, and F. Puthoff which contributed to the research described herein.

REFERENCES

Balci, O. (1986), "Requirements for Model Development Environments," *Computers and Operations Research 13*, 1, pp. 53-67.

Balci, O. and R.E. Nance (1987a), "Simulation Model Development Environments: A Research Prototype," *Journal of the Operational Research Society 38*, 8 (Aug.), 753-763.

Balci, O. and R.E. Nance (1987b), "Simulation Support: Prototyping the Automation-Based Paradigm," In *Proceedings of the 1987 Winter Simulation Conference* (Atlanta, Ga., Dec. 14-16). IEEE, Piscataway, N.J., pp. 495-502.

Balci, O., R.E. Nance, E.J. Derrick, E.H. Page, and J.L. Bishop (1990), "Model Generation Issues in a Simulation Support Environment," In *Proceedings of the 1990 Winter Simulation Conference* (New Orleans, LA, Dec. 9-12). IEEE, Piscataway, N.J., pp. 257-263.

Bishop, J.L. and O. Balci (1990), "General Purpose Visual Simulation System: A Functional Description," In *Proceedings of the 1990 Winter Simulation Conference* (New Orleans, LA, Dec. 9-12). IEEE, Piscataway, N.J., pp. 504-512.

Frankel, V. and O. Balci (1989), "An On-Line Assistance System for the Simulation Model Development Environment," *International Journal of Man-Machine Studies 31*, pp. 699-716.

Nielsen, J. (1990), "Traditional Dialogue Design Applied to Modern User Interfaces", *Communications of the ACM 33*, 10 (Oct.), pp. 109-118.

Sun Microsystems (1986), *SunINGRES*, Volumes I, II, and III, Sun Microsystems, Inc., Mountain View, Calif., May.

Sun Microsystems (1988), *SunView Programmer's Guide*, Sun Microsystems, Inc., Mountain View, Calif., May.