ABSTRACT

The Form 90-A, adapted from the Form 90 developed for Project MATCH (1993), is an interviewer-completed, calendar-based survey instrument designed to assess client alcohol consumption. The SAS/AF FRAME environment of the SAS system was used to develop a graphical user interface for data entry, using object-oriented techniques to facilitate development and modification time. Due to the level of complexity involved, development of the calendar structure proved to tax the abilities of the SAS/AF environment. The calendar consists of thirty-seven Container Boxes with Graphic Texts. Limiting the boxes to display only the dates was necessary to allow the application to run; the difficulty was then in programming, since Test AF was not able to execute the application while PROC BUILD was active. Initial testing with real situations have shown SAS/AF to be an effective data entry solution. Users have reported a vastly improved ease of data entry. Moreover, since all of the calculation work is handled by the computer and unusual, inconsistent, or improper values are reported or prevented, we expect a much more reliable data set to result. Users feel more comfortable interacting in this environment and appreciate seeing their data entered and displayed in a natural manner. Additionally, modifications and improvements have been suggested that were easy to implement in the SAS/AF environment.

INTRODUCTION

Background

The Form 90-A is a structured interview for obtaining information from clients about their alcohol use over roughly the previous 90-day period. Clients return for follow-up interviews periodically so that their daily drinking behavior over the course of a year or more is compiled. Information collected about each day are: What type of drink, How much was consumed, and for how long did they drink, as well as where they resided on that day (home, detox, incarcerated, etc.). From this information a measure of their peak Blood Alcohol Content (BAC) and Standard Ethanol Content (SEC) can be calculated. The information is collected on printed calendars, one for each month in the given 90-day period. Previous experience with the data entry of this form was greatly complicated by the fact that the data entry screens (usually just text lines) had little resemblance to the calendar forms. Further, labor-intensive, individual hand calculations were required to derive the values of BAC and SEC, since the data entry software previously in use was not capable of dealing with data such as alcohol type, amounts etc. What was desired was a data entry screen that was similar to the actual forms used which was also capable of capturing all data collected during the interview. This would have the advantage of decreasing data entry fatigue and error as well as allowing for more detailed investigations of the data set.

Why SAS/AF

The FRAME environment of SAS was chosen since it allows a graphical interface for data entry and would hopefully simplify the programming through its use of Object Oriented Programming. Since the interview sessions often cover four or more months it was quickly realized that displaying data for all the days at once would be prohibitive on the basis of screen clutter alone. Instead, users enter data one month at a time, the month being selected from a List Box. This allows the creation of a single calendar prototype in the Calendar Frame which can be modified to display a given month. The background colors of the calendar days reflect the status of data entry for each particular day. Users click on a day in the calendar and the data entry form for that day appears. This method is employed throughout the screen; objects are only visible when in the correct context, thus reducing entry errors. Data for the drinking on the selected day is entered and the values of BAC and SEC are automatically calculated and displayed. A database of alcoholic beverages is available in a list box, enabling users to select a specific drink. The software then calculates percent alcohol for that drink from the information contained in the database. When the data for the day are complete and another day is selected (clicked on ) from the calendar, the color for the completed data box changes to reflect its new status.

In addition to the calendar information, the Form 90-A also contains roughly fifty questions with numeric answers. A separate form was created listing the questions and Text Entry fields following them. This Form Frame and the Calendar Frame are accessed from a Client Frame (Figure 1) which lists all clients entered in a List Box and allows for the creation and removal of clients as well.

DATA ON DISPLAY

To access the Calendar Frame or the Form Frame, a user must select or enter a client on the Client Frame. The Client Frame takes care of housekeeping tasks such as creating a new client or deleting an old one. Once a client is selected the user can move to the Form Frame or into the Calendar Frame at the click of a button.
The Form Frame (Figure 2) is a representation of the Form 90-A general questions asked at the interview.

After completing the Form Frame, the user clicks on the button labeled "DONE" and is returned to the Client Frame, where the user can choose the "CALENDAR" button to move into the initial Calendar Frame. After inputting the client's weight and date of first drink, the program automatically calculates the number of days or months that need to be entered. A List Box containing valid entry months is then drawn on the screen (Figure 3). Upon choosing a particular month, the second step of the Calendar Frame is then created (Figure 4).

The second step of the Calendar Frame presents the days of the month in a calendar-like format, in which the user can click on a given day to enter data. When the particular day is chosen, the full Calendar Frame is then drawn (Figure 5). This screen allows the user to enter the specific drinking information for the selected day, that is, what type of drink, how much was consumed, and for how long did they drink. Data entry is both accelerated and made easier with the "point and click" interface made available by FRAME. For example, entering the type of drink is facilitated by a list box containing numerous popular drinks as well as various local favorites. This information is contained in a separate SAS data set which also contains the predetermined ethanol content of each type of drink. Since the data set is maintained externally from the program, it is easily updated with new information if the need arises. Living status codes which are linked to each specific day are displayed in a Radio Box, enabling the user to enter that data with the click of a button. Another feature used to simplify data entry is the Weekly Patterns Box. A noticeable trait of problem drinking found in our research is the existence of patterns of drinking; the Weekly Patterns Box enables the data entry clerk to save up to nine different weekly patterns. These saved patterns can be pasted into other weeks, thereby greatly reducing data entry time. BACs and SECs are automatically calculated and displayed in a box in the lower right hand corner of the screen for the selected day. Also available here is the "Abstained" check box, which indicated that the client was abstinent from drinking for the given day.

Data Sets

With screens available that resembled the actual instrument, the data sets had to be defined. Since each client has data for over 365 days and each day may have several drinks in several episodes,
the amount of information being collected can get quite large, making data management cumbersome. Our solution was to create one Output SAS data set that contained only the date, BAC, SEC, and living code for each day, allowing for 365 days before the first interview and up to 10,000 days afterwards. Since the actual period covered is usually much smaller (up to 270 days before and 450 after) the data set is often truncated past variables holding follow up days 450 to 10,000 or as necessary. This cuts the number of variables down to approximately 2000 rather than the 40,000 necessary for a complete data set. The data regarding drink types, amounts, and living status are stored in separate SAS data sets with each record holding one item. For example, if Client C had three different drinks on day N, the beverage set would have three records with ID=C, DATE=N but each holding the data for the respective drinks. This method is much more efficient in terms of storage. The SCL takes care of managing the data sets to present the data in an understandable form on the FRAME screen.
Connecting The Data To The Screens

The following code gives an example of the SQL used to drive the screens. This is the code connected to the CLIENT FRAME.

```sql
-- Version 1.1.1

Written by: Eric Jaderlund
eric@jaderlund.com

© 1994 E. Jaderlund, and M.J. O'Meara III

Version History

1.0.0 Released 11-7-94
First Version to include this screen and Form Screen

1.1.1 Released 11-18-94
Changed to new file naming scheme

1.1.0 Released 2-1-95
New creates new small work datasets for the calendar program to work with to improve speed.

1.1.1 Released 2-2-95
Minor bug fixes

Description

The startup frame for Form90-A. Displays, adds and deletes Form90-A clients and calls the Calendar and Form data entry screens.

EDIT:

FALSE = 0; TRUE = 1;
call MARKE('Form90-A Clients');
link CYCLIS;
call notify('NEWDATA', '_NONE_', 'ALL');
return;

MAIN:
call notify('CLIENTS', '_GET_LAST_SEL_', anySelected);
if ( anySelected = FALSE ) then do;
call notify('DELETE', '_GRAY_');
call notify('EDITFORM', '_GRAY_');
end;
else do;
call notify('DELETE', '_UNGRAY_');
call notify('EDITFORM', '_UNGRAY_');
end;
return;

TERM:
return;

EDITCAL:
set rdadata.F90asehp;
if ( RANKID = 'aseID' ) then DELETE;
data XXXall:
set rdadata.F90asehp;
if ( RANKID = 'aseID' ) then DELETE;
data XXXaut;
set rdadata.F90asehp;
if ( ID = 'aseID' ) then DELETE;
data XXXend;
set rdadata.F90asehp;
if ( ID = 'aseID' ) then DELETE;
run:
endsubmit;
call display( 'masters.frames.F90a_cal.frame', mode, selID, selType );
if ( mode = 2 ) then call display( 'masters.frames.F90a.sav.frame', selID );
call notify('CLIENTS', '_DESELECT_');
return;

EDITFORM:
set rdadata.F90asehp;
if ( RANKID = 'aseID' ) then DELETE;
data XXXall:
set rdadata.F90asehp;
if ( RANKID = 'aseID' ) then DELETE;
data XXXaut;
set rdadata.F90asehp;
if ( ID = 'aseID' ) then DELETE;
data XXXend:
set rdadata.F90asehp;
if ( ID = 'aseID' ) then DELETE;
run:
endsubmit;
call display( 'masters.frames.F90a_cal.frame', mode, selID, selType );
if ( mode = 2 ) then call display( 'masters.frames.F90a.sav.frame', selID );
call notify('CLIENTS', '_DESELECT_');
return;

NEW:
newID = 'BLANK';
newType = ' '; newSex = ' '; call notify('NEWDATA', '_DELETE_ALL_');
return;

ADD:
if ( (newID = 'BLANK') OR (newSex = ' ') ) then do;
alarm;
return;
end;
subFile = open('rdadata.F90asehp', 'U');
call set(subFile);
ok = READ(subFile);
ok = RETURN(subFile);
ok = READ(subFile);
ok = REMove(subFile, 'subID');
ok = FLETCH(subFile, 'subID');
ok = UPDATE(subFile); else do;
call display( 'masters.frames.F90a_cal.frame', '
Sorry, that ID already exists!',

/}
```

-- Version 1.1.1

Written by: Eric Jaderlund
eric@jaderlund.com

© 1994 E. Jaderlund, and M.J. O'Meara III

Version History

1.0.0 Released 11-7-94
First Version to include this screen and Form Screen

1.1.1 Released 11-18-94
Changed to new file naming scheme

1.1.0 Released 2-1-95
New creates new small work datasets for the calendar program to work with to improve speed.

1.1.1 Released 2-2-95
Minor bug fixes

Description

The startup frame for Form90-A. Displays, adds and deletes Form90-A clients and calls the Calendar and Form data entry screens.

EDIT:

FALSE = 0; TRUE = 1;
call MARKE('Form90-A Clients');
link CYCLIS;
call notify('NEWDATA', '_NONE_', 'ALL');
return;

MAIN:
call notify('CLIENTS', '_GET_LAST_SEL_', anySelected);
if ( anySelected = FALSE ) then do;
call notify('DELETE', '_GRAY_');
call notify('EDITFORM', '_GRAY_');
end;
else do;
call notify('DELETE', '_UNGRAY_');
call notify('EDITFORM', '_UNGRAY_');
end;
return;

TERM:
return;

EDITCAL:
set rdadata.F90asehp;
if ( RANKID = 'aseID' ) then DELETE;
data XXXall:
set rdadata.F90asehp;
if ( RANKID = 'aseID' ) then DELETE;
data XXXaut;
set rdadata.F90asehp;
if ( ID = 'aseID' ) then DELETE;
data XXXend:
set rdadata.F90asehp;
if ( ID = 'aseID' ) then DELETE;
run:
endsubmit;
call display( 'masters.frames.F90a_cal.frame', mode, selID, selType );
if ( mode = 2 ) then call display( 'masters.frames.F90a.sav.frame', selID );
call notify('CLIENTS', '_DESELECT_');
return;

EDITFORM:
set rdadata.F90asehp;
if ( RANKID = 'aseID' ) then DELETE;
data XXXall:
set rdadata.F90asehp;
if ( RANKID = 'aseID' ) then DELETE;
data XXXaut;
set rdadata.F90asehp;
if ( ID = 'aseID' ) then DELETE;
data XXXend:
set rdadata.F90asehp;
if ( ID = 'aseID' ) then DELETE;
run:
endsubmit;
call display( 'masters.frames.F90a_cal.frame', mode, selID, selType );
if ( mode = 2 ) then call display( 'masters.frames.F90a.sav.frame', selID );
call notify('CLIENTS', '_DESELECT_');
return;

NEW:
newID = 'BLANK';
newType = ' '; newSex = ' '; call notify('NEWDATA', '_DELETE_ALL_');
return;

ADD:
if ( (newID = 'BLANK') OR (newSex = ' ') ) then do;
alarm;
return;
end;
subFile = open('rdadata.F90asehp', 'U');
call set(subFile);
ok = READ(subFile);
ok = RETURN(subFile);
ok = READ(subFile);
ok = REMove(subFile, 'subID');
ok = FLETCH(subFile, 'subID');
ok = UPDATE(subFile); else do;
call display( 'masters.frames.F90a_cal.frame', '
Sorry, that ID already exists!',
Development of the calendar structure proved to tax the abilities of the FRAME/AF environment. The calendar consists of thirty seven Container Boxes with Graphic Text displaying the dates. Originally, it was hoped to have more items in each Date Box, but this large number of graphic objects already exceeded the memory management capabilities of SAS-PC. This difficulty was corrected by displaying only the dates in the boxes. Programming also proved to be troublesome due to the inability of Test AF to run while a frame is active, namely having to exit the build frame routine and execute it to start and redrawing the calendar can take almost as long. This delay is barely tolerable in our data entry environment. Also, FRAME redraws all objects on the screen after any object is updated. There were also difficulties with Hot Spot objects interacting in odd ways with Text Entry objects, and also problems getting Extended Tables to load and refresh properly.

RESULTS

The initial tests with data entry in the FRAME environment have been extremely encouraging. Ease of data entry has greatly improved, and a higher level of data reliability is anticipated. The software presents a screen that looks similar to the paper instrument and is capable of tracking an individual's drinking on a daily basis. The one major limitation is hardware requirement. A minimum of a 486-66 DX2 with 16 MB of memory is needed for the...
application to run adequately, but a 90 MHz Pentium with 16 MB is recommended. Software requirements are Windows version 3.1 or above and SAS for Windows version 6.08 or above including the BASE module.

The Form 90-A data entry system was developed over a period of three months, utilizing approximately 160 programming and development hours. The new system has decreased data entry time by over 30%, which resulted in an estimated yearly savings of 480 hours or $4500. The total for development of the Form 90-A program was under $2500, which confirmed our expectations that SAS would improve the overall cost-effectiveness of the data entry department.

SAS and SAS/AF are registered trademarks of SAS Institute Inc. In the USA and other countries. ® indicates USA registration.