ABSTRACT

Throughout the automotive direct marketing industry there is increasing pressure to produce high quality research under very tight time and cost constraints. Out of these concerns arose the need for an interactive computer system which could perform the myriad tasks associated with automotive research yet still meet strict time and cost constraints. The system developed is known as ARAS (Automotive Research Analysis System) and it addresses the above concerns by providing an online environment which automates many of the routine functions associated with automotive research.

INTRODUCTION

The task of automotive research at R. L. Polk is to assist clients in selecting the best prospective customers for their goods or services. Within research there are three primary methods to accomplish this task. The first technique involves development of a statistical model which identifies a subset of prospective customers from a large national database. This process involves building an analysis file, applying custom selection factors, producing a model of key predictor characteristics and then lastly, comparing the output generated to a control file. The overall process requires extreme flexibility as each step will differ from project to project. A second type of analysis is to determine the ownership base for certain groups of car purchasers. For example, one might be interested in determining the types of vehicles owned by persons purchasing Ford Escorts®. A final type of analysis is to investigate the key demographic characteristics for a selected purchase group. For example, one might be interested in the average number of children within households that purchased Chevrolet Berutas®. If this number was high then perhaps Chevrolet should target households with children as prospective clients.

For the above types of research, client requirements will differ from project to project and thus each project was done on a semi-custom basis. This style of research was effective in meeting client needs but was also costly and time consuming. To eliminate these problems the ARAS system was designed and implemented. SAS Display Manager was chosen as the development tool as it was familiar to our user community and offered an easy to use windowing environment.

REQUIREMENTS

The system was designed to meet the following requirements:

1. Provide complete automation for 60% - 70% of Automotive Research projects.
2. Produce a 30% or better reduction in average project cost.
3. Reduce average project turnaround time by 50%.
4. Provide ability to process a SAS dataset containing up to 1.5 million observations and 200 variables in an IBM/TSO® environment.
5. Provide capability to add new components without making major changes to the existing system.
6. Provide a consistent, easy to use interface that can be utilized by persons who are not SAS programmers.
7. Allow for up to 8 concurrent users.

DEVELOPMENT

Because ARAS is designed to accomplish a number of distinct research tasks it was decided to create a series of subsystems, all of which are linked together in a SAS windowing environment. The subsystems are accessed from a common main menu and each subsystem returns to this menu when its task is completed.

Data linkage between subsystems is accomplished through the use of a working dataset. This dataset is created by processing user input and will change as the user executes the various subsystems of ARAS. The name of the working dataset is stored in a macro variable which is available to all the ARAS subsystems.

The overall structure of each subsystem is very similar. Each consists of two SAS programs. The first is a screen program and the second is a processing program. The screen program displays various SAS windows and collects user input. This input is then stored in a series of macro variables.
The screen program within each subsystem is also responsible for validation of user requests. To assure a high degree of accuracy, validation is done at three levels. At the lowest level is a check that required input has been entered and that the input is within acceptable ranges. At the next level, each window offers action bar options which allow the user to redo what was entered or return to the main menu without executing the processing section of the subsystem. At the highest level, the screen program will produce a summary file which contains a record of all processing steps.

The second program performs actual processing by converting macro variables (created in the screen program) into actual SAS code which is then executed. Outputs produced include reports, SAS datasets, and audit or summary files.

Within ARAS users have great flexibility in choosing any of 175 different analysis variables. To handle this level of flexibility requires that each processing program be heavily macro oriented. The majority of code in the processing programs is created by executing macros which in turn generate SAS code. Below is an example macro from one of the ARAS subsystems. This particular macro generates SAS code which will delete certain analysis records based on household vehicle ownership. The macro variables used are global in scope and are generated by the separate screen program.

```
%macro compdel;
* SET UP COMPETITIVE DELETIONS FOR OWNED FIELDS *
%if "&01" = "I" OR "&02" = "I" OR "&03" = "I" OR "&04" = "I" OR "&05" = "I" %then %do;
  if substr(oln1,l,3) = substr("OWN",1,3) or substr(oln1,l,3) = substr("OWN",1,3) %then %do;
    if substr(oln1,l,4) = substr("OWN",1,4) or substr(oln2,l,4) = substr("OWN",1,4) %then %do;
      %end;
    %end;
  %end;
%end;
```

The figure below represents the overall system model for ARAS. The initial database used can be either a predefined household dataset or a user defined dataset. The selected database can be accessed by any or all of the ARAS subsystems.

The current ARAS subsystems perform the following operations:

**FILE SELECTION:** Prompts the user to identify a dataset which will be the starting point for analysis. The user can choose from a default master database, portions of that database or a user supplied database. The database chosen becomes the initial working dataset for ARAS.

**CUTBACK:** Selects a specific analysis population from the working dataset. Selection criteria include: age of vehicles owned; type of vehicles owned; number of vehicle purchases; as well as numerous other criteria. Any combination of selection criteria may be utilized.

**SPECIAL VARIABLES:** Allows the user to create up to 82 additional analysis variables. These variables are specific to a particular project and are not stored on the initial master database. Examples include, the number of luxury cars owned, the number of domestic cars owned, or the number of previous new car purchases.

**LOGIT MODEL:** After the user has defined an analysis file using the Cutback and Special Variable subsystems the Logit subsystem may be run. This series of programs uses custom SAS procedures to develop a model of purchasing behavior.
SINGLE FACTORS: Performs a demographic analysis of a particular purchasing group. This subsystem answers questions such as: What is the age distribution for Chevrolet Firebird purchasers? Currently, over 175 variables (which can be chosen in any combination) are available for analysis.

BUY RATE: Uses the current working dataset to analyze and report the ownership base for a user defined group of car purchasers. This subsystem can answer questions such as: What type of vehicles are most likely to be owned by persons purchasing some particular car, such as a Ford Escort?

OUTPUT: Takes the current working dataset and converts it to a permanent SAS dataset which is then stored on disk under a name provided by the user. Because this can be a time consuming process, a counter indicating records processed is displayed on the users terminal.

EFFICIENCY
Since ARAS is used as an online operational environment for an entire department, it is extremely important for programs to be fast and efficient. Some of the techniques used to accomplish these goals include:

1. Use of SAS macros to generate only code that was absolutely necessary. Since the choice of analysis variables is up to the user, it is critical that only code relevant to selected variables is generated.

2. Use of built-in sampling routines which help to reduce dataset observations. ARAS internally calculates optimal sampling sizes but this can be overridden by the user.

3. Use of a Cutback Subsystem to reduce analysis population size before performing major processing.

4. Use of Length statements to reduce the size of numeric variables.

5. Use of Keep statements so that only required variables are processed and kept on datasets.

6. Use of PUT and INPUT functions to avoid automatic SAS controlled numeric or character conversions.

7. Use of PROC FREQ and PROC SUMMARY to reduce dataset observations before intensive processing phases.

8. Use of the ‘BUFNO’ and ‘BUFSIZE’ dataset options. When using these options response time improvement was 30%.

9. Use of strict data validation to assure user input will accomplish the desired task.

OPERATION
ARAS is a large system which uses close to 100 screens; therefore, it is not feasible to show all the possible processing options in this paper. However, to develop a good understanding of ARAS operations, key screens may be reviewed. The example which follows shows a typical session when using the Cutback subsystem.

Screen 1. This is the main menu for ARAS. This screen acts as a gateway for entering into any of the ARAS subsystems. Also, when processing for a subsystem concludes, the user is returned to this screen. For this example, assume the user chooses to enter the Cutback Subsystem (Note: this screen is an approximation of the actual ARAS main menu).

Screen 2. This screen allows the user to select the year model range for owned vehicles. The screen provides a default range of 7 years but any range from 00 to the current year can be chosen. In this case the user presses the enter key to accept the default. Also, on this screen one should take note of the action bar options REDO and CANCEL. These are provided on most ARAS screens and give the choices of re-entering input (REDO) or returning to the main menu (CANCEL).
Screen 3. This screen is used to select households that own vehicles purchased new, used or both. The screen provides a default entry of 'BOUGHT NEW' which is selected by the user. (Note: default entries are selected when the user presses the enter key without typing in any values).

Screen 4. This screen allows for the selection of analysis states. For maximum flexibility the user can select states to include or states to delete. Here the user selects state inclusion.

Screen 5. A list of potential states is then displayed and the user selects those to be included in the analysis. A selection is made by placing an 'x' next to the state of interest.

Screen 6. This screen allows the user to choose up to four additional deletion or inclusion options. These options provide for an analysis of customer purchasing behavior over time. In this case the user presses enter to bypass these options.
Screen 7. This screen is used to determine if a dependent variable is to be coded. A dependent variable is an overall factor which defines a group of interest. For this example the user selects 'YES'.

Screen 8. In this screen the user selects criteria for determining the dependent variable. Here the user can select to use vehicle codes or enter actual SAS code. In this case the user chooses to enter SAS code.

Screen 9. In this screen the user enters code which will represent the condition portion of an If statement. This code will be inserted into the processing program.
Screen 12. After the cutback program runs a summary screen is displayed. This screen provides an option to sample the newly created analysis file using default or user defined factors. Also, the user is informed that an audit file containing all cutback selections and counts has been created. In this case the user decides to sample.

Screen 13. After the sampling process concludes a screen showing final results is displayed for the user. At this point the user presses the enter key and is returned to the main menu. Once back at the main menu the user can exit ARAS or proceed to run another subsystem.

CONCLUSIONS

The implementation of ARAS has been extremely successful for the Automotive Research department at R.L. POLK. In fact, the system has met or exceeded virtually all of the requirements defined above. To illustrate, in the pre ARAS era 30% - 40% of total project cost was incurred in developing and executing semi-custom batch programs. Also, the coding process would normally take 5 - 7 work days. With the interactive ARAS environment all batch program development costs have been eliminated and project turnaround time has gone from 5 - 7 days to 1 - 2 days.

One other unexpected benefit is that the training of research analysts has been greatly simplified. With the common consistent interface of ARAS, new research analysts become productive much more quickly and make fewer mistakes.

To sum up, the creation of ARAS has led to productivity gains that were not thought to be possible. In the future this success story should continue as plans are already underway to add five new subsystems to ARAS.

ACKNOWLEDGMENTS

The author would like to thank Steve Caldwell and Chris Hurley for their design and programming efforts in building the ARAS system.

SAS is a registered trademark or trademark of SAS Institute Inc., in the USA and other countries. ® indicates USA registration.

IBM and TSO are registered trademarks or trademarks of International Business Machines Corporation. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT

For more information please contact:

Michael Sternberg
Manager - Automotive Research Programming
R.L. POLK
2736 Pembroke
Birmingham, MI 48009