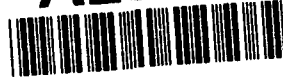


AL-TP-1991-0009

AD-A236 033



THE INTEGRATED DECISION MODELING SYSTEM (IDMS) USER'S MANUAL

Jonathan C. Fast  
John N. Taylor

Metrica, Incorporated  
8301 Broadway, Suite 215  
San Antonio, TX 78209

Larry T. Looper

HUMAN RESOURCES DIRECTORATE  
MANPOWER AND PERSONNEL DIVISION  
Brooks Air Force Base, TX 78235-5000

DTIC  
ELECTE  
MAY 14 1991  
S B D

May 1991

Final Technical Paper for Period March 1989 - February 1991

Approved for public release; distribution is unlimited.

DTIC FILE COPY

01 5 14 013

AIR FORCE SYSTEMS COMMAND  
BROOKS AIR FORCE BASE, TEXAS 78235-5000

ARMSTRONG LABORATORY

## NOTICES

This technical paper is published as received and has not been edited by the technical editing staff of the Armstrong Laboratory.

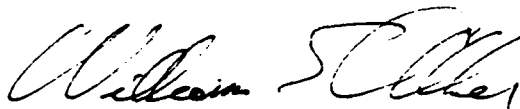
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Office of Public Affairs has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.



LARRY T. LOOPER  
Project Scientist



WILLIAM E. ALLEY, Technical Director  
Manpower and Personnel Division



MICHAEL W. BIRDLEBOUGH, Colonel, USAF  
Chief, Manpower and Personnel Division

**REPORT DOCUMENTATION PAGE**Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> May 1991	<b>3. REPORT TYPE AND DATES COVERED</b> Final Paper - March 1989 - February 1991	
<b>4. TITLE AND SUBTITLE</b> The Integrated Decision Modeling System (IDMS) User's Manual			<b>5. FUNDING NUMBERS</b> C - F41689-88-D-0251 PE - 62205F PR - 7719 TA - 20 WU - 20	
<b>6. AUTHOR(S)</b> Jonathan C. Fast John N. Taylor Larry T. Looper				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Metrica, Incorporated 8301 Broadway, Suite 215 San Antonio, TX 78209			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)</b> Armstrong Laboratory Human Resources Directorate Manpower and Personnel Division Brooks Air Force Base, TX 78235-5000			<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b> AL-TP-1991-0009	
<b>11. SUPPLEMENTARY NOTES</b>				
<b>12a. DISTRIBUTION/AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited.			<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (Maximum 200 words)</b>  The Integrated Decision Modeling System (IDMS) is a prototype computer-based decision aid which incorporates a user interface to help the decision maker analyze a problem, determine the most appropriate decision aiding tool, and utilize the selected tool. Each of the four decision aiding models as well as the user interface and sensitivity analysis modules are presented in appendices which discuss the technique, provide examples of their use, and instruct the user as to how to access each through the computer.				
<b>14. SUBJECT TERMS</b> decision analysis decision modeling			hierarchical decision models multiattribute utility	policy capturing
			<b>15. NUMBER OF PAGES</b> 108	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UL	

## TABLE OF CONTENTS

	Page
I. INTRODUCTION .....	1
Description of Techniques .....	1
II. THE INTEGRATED DECISION MODELING SYSTEM ...	3
Technique Selection Module .....	3
Problem Structuring Module .....	3
Sensitivity Analysis Module .....	6
REFERENCES .....	7
APPENDIX A: POLICY SPECIFYING PROGRAM (POLSPEC) USER'S MANUAL RELEASE 2.1 .....	9
APPENDIX B: POLICY CAPTURING PROGRAM (POLCAP) USER'S MANUAL RELEASE 1.0 .....	38
APPENDIX C: SIMPLE MULTIATTRIBUTE RATING TECHNIQUE (SMART) USER'S MANUAL RELEASE 1.0 ....	53
APPENDIX D: HIERARCHICAL ADDITIVE WEIGHTING METHOD (HAWM) USER'S MANUAL RELEASE 1.0 .....	75
APPENDIX E: SENSITIVITY ANALYSIS MODULE USER'S MANUAL RELEASE 1.0 .....	83



<b>Accession For</b>	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
<b>Availability Codes</b>	
Dist	Avail and/or Special
A-1	

## LIST OF FIGURES

Figure		Page
1	User's Interface Screen .....	4
2	Payoff of Context/Technique Match .....	5

## LIST OF TABLES

Table		Page
1	Attribute Listing .....	5

## PREFACE

The Integrated Decision Modeling System (IDMS) is the first step in the development of a decision analysis and policy modeling capability to assist Air Force managers in making effective use of scarce resources through analysis of resource trade-offs. This work is part of the Manpower and Personnel Division's decision modeling research program.

## SUMMARY

A prototype integrated decision modeling system (IDMS) was developed to assist Air Force manpower and personnel planners in utilizing decision analysis techniques. IDMS has modules for technique selection, problem structuring, policy development, and sensitivity analysis. Prototype versions of all these modules were developed and incorporated in the IDMS. The technique selection module assists the user in selecting a decision modeling technique for a specific decision context. The module will request information on 15 context attributes, compute a utility for each technique, and then display these utilities so that the user can determine which technique is best suited for this decision context. The problem structuring module enables the user to build a hierarchy for analyzing a decision problem. This hierarchy can be used as the input to several decision modeling techniques. The policy development module allows access to four techniques that can be used for decision modeling. These four techniques are: policy specifying, policy capturing, simple multiattribute rating technique (SMART), and hierarchical additive weighting method (HAWM). The final module performs sensitivity analysis with the use of a program called payoff generator (PAYGEN).

## I. INTRODUCTION

Scientists at the Air Force Human Resources Laboratory (AFHRL) have been researching and applying decision analysis tools in a wide variety of decision contexts since the early 1960's. Recently AFHRL completed a research program to enhance the laboratory's decision analysis and policy modeling capability (Fast & Looper, 1989). This present research effort provides a detailed users manual for a decision support system which incorporates four decision modeling techniques into a single prototype software package, the Integrated Decision Modeling System (IDMS). This software package determines the utility of applying each of the four decision modeling techniques to any resource allocation decision context. The four techniques in IDMS include: policy specifying, policy capturing, simple multiattribute rating technique (SMART), and hierarchical additive weighting method (HAWM). These four techniques are described briefly in this document. For a more detailed discussion of the four techniques and the evaluation of each technique in three different Air Force decision contexts, see Fast & Looper, 1989.

### Description of Techniques

Policy specifying was developed by AFHRL (Ward, 1977). A user's manual for this technique is found in Appendix A to this report. The primary motivation behind its development was to allow a decision maker to express mathematically a decision policy function which permits interactions among decision attributes which other techniques were unable to accommodate readily. In policy specifying, the analyst aids the decision maker in defining the problem, to include determining the decision objective and the decision attributes. These attributes are then built into a hierarchy, where attributes can be interacted pairwise to form a single measure of utility (or payoff). The policy specifying procedure continues with the determination of the functions that describe each pairwise interaction in the hierarchy. The decision maker is asked to specify payoff values for endpoint combinations of attribute pairs and the slopes and inflections in one of two starting models (Ward, 1977). When these specifications or restrictions are applied to the starting model, the unknown weights in the model can be solved.

The process is iterative in that the decision maker reviews the pairwise payoff table that results from the mathematical model to determine if the desired interaction has been specified. Changes can be made in the slopes and inflections or in the restrictions so that the resulting payoff table more accurately reflects the decision maker's policy. This process is continued until each of the pairwise functions that make up the hierarchy have been determined. Research in this project suggested that the pairwise function could in some cases be more easily determined by stating the numerical payoff table first, and then determining the function that best fits this table. This capability was included in the software package as an alternative methodology.

The second technique is often called policy capturing but is also known by other names such as judgment analysis (Christal, 1963). The user's manual for this technique is found in Appendix B to this report. Versions of the technique have been widely used in many contexts (Hammond, Stewart, Brehmer, & Steinmann, 1975). Policy capturing is a holistic technique that attempts to replicate the observed decisions. The analyst must develop a representative decision set which is presented to a judge (or judges) holistically. The judges rank order or score each item in the set to provide numerical judgment values that can be analyzed mathematically. To complete the process of policy capturing, linear regression is used to regress the attributes, using the rank order or score as the dependent variable. The resulting equation represents the judge's policy, "capturing" the observed decision process. In the case of multiple judges, a clustering technique may be used to determine the single equation that best represents the group decision process (Christal, 1967).

The third technique, SMART, is a multiattribute utility theory technique developed by Edwards (1971). A user's manual for this technique is found in Appendix C to this report. In the swing weighting implementation of SMART in IDMS the decision maker is asked to develop a hierarchy that represents the decision problem as in policy specifying, but the hierarchy can be more general than the pairwise construction of policy specifying. The decision maker is then asked to use the swing weighting methodology (Von Winterfeldt and Edwards, 1986), which elicits ratings for a full "swing" (from worst to best) in one attribute while holding the others constant, at each level of the hierarchy. In addition the twigs (or ends) of the hierarchy are examined to determine the relative utilities associated with each of the attributes at the ends. A simple weighted additive function is then used to determine the rank order of each decision alternative as it is presented.

The fourth technique, HAWM, was developed by Hwang and Yoon (1981), based on research by Saaty (1980). A user's manual for this technique is found in Appendix D to this report. HAWM is a software implementation of Saaty's analytical hierarchy process. HAWM uses a hierarchy to perform the decision modeling, although the process is somewhat different from the other hierarchical processes. The hierarchy consists of a top level or decision objective, and a bottom level which consists of the decision options. In between there can be several levels which contain attributes in the decision context. The hierarchy is divided into value trees, and a pairwise comparison of all attributes at each intermediate level is carried out. This pairwise comparison is done to determine the relative importance of the two attributes. At the bottom level of the hierarchy a determination is made of how much of each attribute each of the decision options possesses relative to all the other decision options. An additive model similar to SMART is then used to rank order the decision options.

## II. THE INTEGRATED DECISION MODELING SYSTEM

The prototype software package developed during this project allows the user access to each of the four techniques. The user gains access to the IDMS through an interface module. In addition to executing each of the four techniques the user has modules available for technique selection, problem structuring, and sensitivity analysis. This is shown in Figure 1, the IDMS user's interface screen.

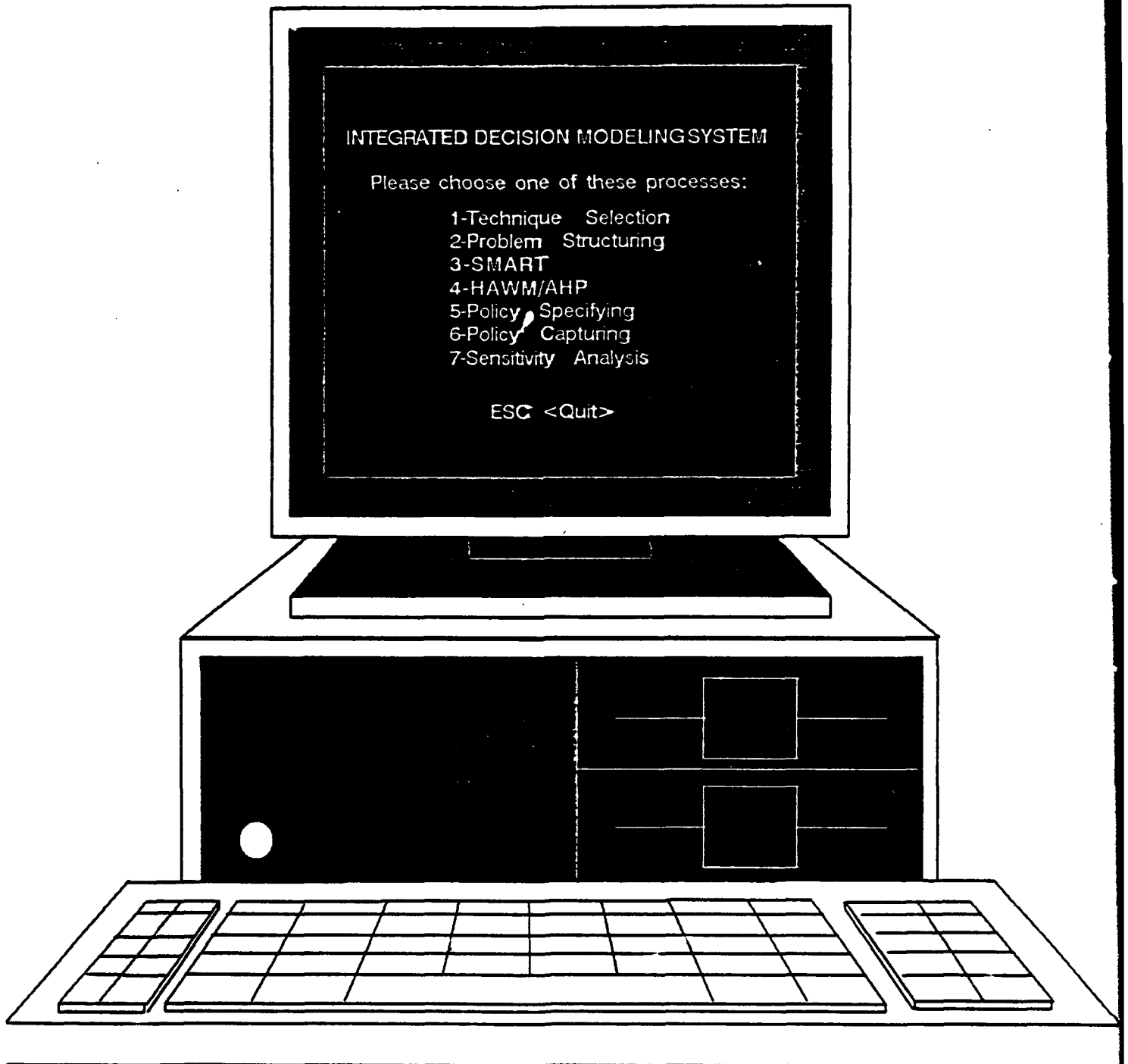
### Technique Selection Module

The technique selection module offers the user help in choosing one of the four techniques for use in a particular decision context. This was done by incorporating a computerized technique/context evaluation module (Fast & Looper, 1989) in the IDMS. The technique selection module contains fifteen attributes or key characteristics of any decision context. These attributes are listed in Table 1. Fast & Looper (1989) developed a rating for each technique's ability to provide each of the fifteen attributes. These ratings are used in the IDMS.

The same fifteen attributes contained in a rating form for decision contexts are incorporated into the technique selection module so that the user can input into a decision analysis program the need for each of the fifteen attributes. The methodology for technique selection is completed by using policy specifying and SMART to determine the utility of matching a context's need for one of the fifteen attributes with the ability of a decision modeling technique to supply that attribute. Policy specifying was used to specify the value of the match between the need for an attribute in the decision context and the ability of the technique to provide the attribute, resulting in the payoff matrix of Figure 2 (Fast & Looper, 1989). The SMART technique is used to elicit the utility of each of the context-technique attribute matches. The overall rank order of the four techniques in a particular decision context is then determined and displayed to the user.

### Problem Structuring Module

In the problem structuring module (PSM) the user is guided through the process of describing the problem context that he/she is studying. The user can specify the attributes of the decision options, and the hierarchy that will be used in one or more of the decision modeling techniques to calculate the utilities of applying the techniques to various decision contexts. In the prototype version incorporated here, the PSM can only be used to describe a hierarchy that will be used with the SMART technique.



**Figure 1. User's Interface Screen**

**Table 1. Attribute Listing**

- |  |   |
|--|---|
| 1. model judgmental dependence                       | 9. produce an acceptable final product                      |
| 2. model decisions holistically                      | 10. be used without a computer                              |
| 3. aid understanding of the problem                  | 11. model a decision environment with many decision options |
| 4. communicate the technical aspects                 | 12. be applied to a new decision option of the problem set  |
| 5. be used with little training                      | 13. be used with little analyst involvement                 |
| 6. develop a theoretically defensible decision model | 14. be used with little decision maker involvement          |
| 7. expand to incorporate new information             | 15. model a group decision making process                   |
| 8. perform sensitivity analysis                      |   |

**Technique Capability**

	1	2	3	4	5	6	7	8
1	100	100	100	100	100	100	100	100
2	64	100	100	100	100	100	100	100
3	53	62	100	100	100	100	100	100
4	42	52	62	100	100	100	100	100
5	32	42	52	63	100	100	100	100
6	21	32	43	54	65	100	100	100
7	11	22	34	45	57	69	100	100
8	0	12	24	37	49	61	73	100

**Figure 2. Payoff of Context/Technique Match.**

### Sensitivity Analysis Module

In the sensitivity analysis module the user can explore the sensitivity of a decision context to changes in the decision model or attribute value. The user is given the opportunity to input a set of decision options, and a decision model that he/she wishes to explore. The model is then applied to the decision options and the results printed for the user. The user can then either alter the set of decision options, or use a different model in which are imbedded slight differences in policy to analyze the sensitivity of the decision context to changes in either decision context or decision model. In the prototype version described here, the user can apply sensitivity analysis to a policy generated using the policy specifying technique. This program is called payoff generator (PAYGEN) and is described in Appendix E to this report.

## REFERENCES

- Christal, R.E. (1963). JAN: A technique for analyzing group judgments (PRL-TDR-63-3). Lackland AFB, TX: Personnel Resources Laboratory.
- Christal, R.E. (1967). Selecting a harem -- and other applications of the policy-capturing model (PRL-TR-67-1, AD-658 025), Lackland AFB, TX: Personnel Resources Laboratory.
- Edwards, W. (1971, June). Social utilities. In Engineering Psychologist, Proceedings of the Symposium on Decision and Risk Analysis Series. U.S. Naval Academy, MD.
- Fast, J.C. & Looper, L.T. (1989). Multiattribute decision modeling techniques: A comparative analysis (AFHRL-TR-88-3). Brooks AFB, TX: Manpower and Personnel Division, Air Force Human Resources Laboratory.
- Gardiner, P., & Edwards, W. (1975). Public values: Multiattribute utility measurement for social decision making. In M.F. Kaplan, & S. Schwartz (Eds.), Human judgment and decision processes. New York: Academic Press.
- Gabrielli, W., & Von Winterfeldt, D. (1978). Are importance weights sensitive to the ranges of alternatives in multiattribute utility measurement? (SSRI-TR-78-6). University of Southern California, CA: Social Science Research Institute.
- Hammond, K.R., Stewart, T.R., Brehmer, B., & Steinmann, D.O. (1975). Social judgment theory. In M.F. Kaplan and S. Schwartz (Eds.), Human judgment and decisions processes. New York: Academic Press.
- Hwang, C., & Yoon, K. (1981). Multiple attribute decision making: Methods and applications. Berlin: Springer-Verlag.
- Raiffa, H. (1969). Preferences for multiattributed alternatives. (RAND Memorandum No. RM-5868-DOT/RC). Santa Monica, CA.
- Saaty, T. (1980). The analytic hierarchy process. New York: McGraw Hill.
- Stillwell, W., Von Winterfeldt, D., & John, R. (1987). Value tree analysis of energy alternatives. Management Science, 33, 442-450.

Von Winterfeldt, D., & Edwards, W. (1986). Decision analysis and behavioral research. Cambridge University Press.

Ward, J.H., Jr. (1977). Creating mathematical models of judgment processes: From policy capturing to policy-specifying (AFHRL-TR-77-47, AD-A048 983). Brooks AFB, TX: Occupation and Manpower Research Division, Air Force Human Resources Laboratory.

**APPENDIX A**

**POLICY SPECIFYING PROGRAM**  
**(POLSPEC)**

**USER'S MANUAL**

**RELEASE 2.1**

## TABLE OF CONTENTS

	Page
I. INTRODUCTION TO POLICY SPECIFYING .....	11
II. GENERAL PRINCIPLES AND FEATURES .....	12
File Structure .....	13
III. USING THE PROGRAM .....	15
Work Option .....	16
File Specification .....	16
Transactions .....	17
Policy Function Output Option .....	29
File Specification .....	29
Designation of Output Device .....	30
Designation of Policy Functions .....	30
Policy Function Output Print Options .....	31
Designation of Data Points for Surface Fitting Equations .....	31
Recovery Option .....	32
Surface Fitting Option .....	32
Error Handling .....	37

## LIST OF TABLES

Table	Page
A-1 Sample Policy Function Hierarchy .....	12
A-2 Sample Policy Function Modification Screen .....	19
A-3 Sample Policy Function Creation Screen .....	19
A-4 Policy Function Output .....	24
A-5 Policy Function Output: Listing of Sorted Payoffs .....	25

## I. INTRODUCTION

Policy specifying (POLSPEC) is a decision modeling technique developed by the Air Force Human Resources Laboratory. Policy specifying enables a decision maker to form a policy which will model his/her decision strategy with respect to a specific problem. The policy specifying process consists of three basic steps:

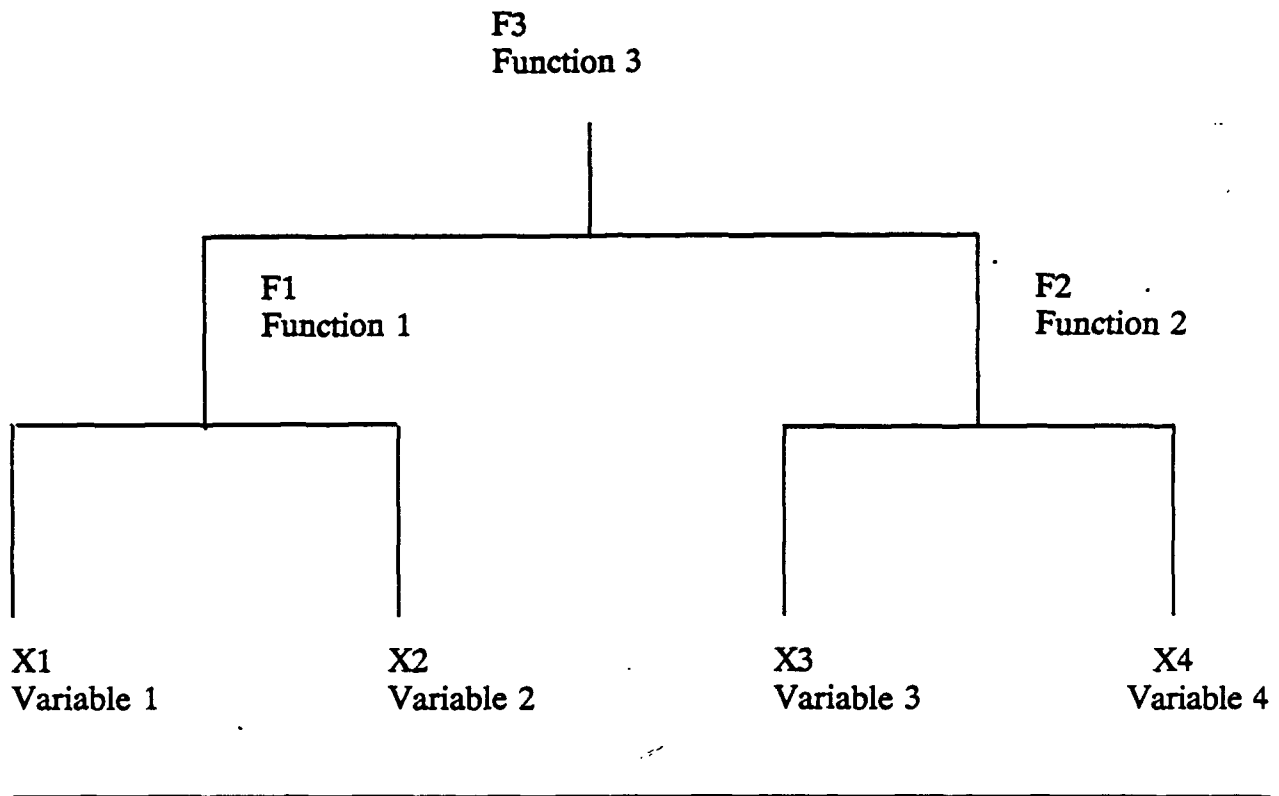
1. Identify the overall objective or criterion of interest and the dimensions or attributes which are important to the problem.
2. Define a pairwise hierarchy of problem attributes and pseudo-attributes which lead up to the final utility or payoff value.
3. Specify each pairwise combination through the implementation of a set of mathematical restrictions on one of two special linear models.

Steps 1 and 2 should be attempted before using POLSPEC. POLSPEC is not designed to accomplish either of these activities; it facilitates the third step.

Policy specifying does not require any special assumptions about the problem attributes or their structuring in the hierarchy. The pairwise comparisons are specified through the application of a set of restrictions on two general linear base models. The development of the policy specifying base models is described by Ward (1977).

A policy specifying hierarchy starts at the lowest level with a set of base variables. These variables combine to form the first level of functions. This is a characteristic of all policy specifying function hierarchies: the lowest level consists of variables and the second level consists of functions of variables. From the third level on up, any function can be a function of variables, a function of lower level functions, or a function of a lower level function and a variable. For the remainder of this document, the term "attribute" will refer to either a function argument which is a variable or a function argument which is a lower level function. A sample policy function hierarchy is shown below in Table A-1.

**Table A-1. Sample Policy Function Hierarchy**



## II. GENERAL PRINCIPLES AND FEATURES

The following "rules of thumb" apply to the operation of the program unless otherwise stated.

1. ALL keyboard entries must be in UPPER CASE
2. Only one keyboard entry may be made at a time
3. ALL program entries except function key entries are input by typing in the appropriate characters followed by the RETURN key
4. Default values can often be entered by typing only the RETURN key
5. Program prompt - User reply transactions are mediated through the use of special prompt screens

6. All of the prompt screens allow for a "help" request for an explanation of what is being prompted. The "F1" function key conveys this request.
7. Some of the prompt screens allow for other special requests which are listed below:

"End"	--> end this prompt session - keep the changes
"RETURN"	--> keep the current value or use the default
"Esc"	--> end this prompt session - cancel this entry

### File Structure

POLSPEC can access different files in the course of its operation. These files are described below.

1. **VARIABLE NAME FILE.** The variable name file is a direct access file that contains the names for the basic variables which form the initial level of a policy function hierarchy. These names are provided by the user during a policy function work session (option "W"). The default extension name for this file is "NAM" and it may contain up to fifty names. Each variable name may be up to 44 characters in length. Each record in the variable name file is made up of two fields. An alphanumeric identifier, four characters in length, is not used by POLSPEC, but exists as an aid in examining the file with the DOS editor. The actual variable name constitutes the second field.

2. **POLICY FUNCTION FILE.** The policy function file is a sequential file which is the permanent repository for a policy. The policy function file contains all of the parameter information for a policy function. This file can hold up to fifty functions. The policy function file is accessed only at the beginning and end of a policy session. For this reason, the policy file can be considered a more "secure" storage medium than the policy work file which is accessed throughout a policy session. The extension name default for this file is "POL".

3. **POLICY WORK FILE.** The policy work file is a direct access file which is used throughout a policy session to read and write policy functions. This file can hold up to fifty functions. Though POLSPEC accesses this file continuously throughout a policy session, the file is opened and closed on each access. This means that the policy work file is relatively secure from program "bomb-out" or a system "crash". The extension name default for this file is "TMP".

4. **POLICY FUNCTION OUTPUT FILE.** The policy function output file is a sequential file into which POLSPEC writes the output of functions designated by the user. This file can be accessed through either the policy work or the policy output run options. If this file is accessed through the work ("W") option, the output of only a single function can be stored there. If the output ("P") option is run, this file will contain output for all of the functions designated by the user. In either case, this output file provides an alternative to the use of the printer as a means of obtaining policy function output. The default extension for this file is "PRN".

5. **GRAPHIC DATA FILE.** The graphic data file can only be accessed during the policy work option ("W"). The purpose of this sequential file is to store the necessary policy attribute and function payoff value information, along with special 3-D plot parameters, for use by any 3-D graphics package which can produce plots of policy functions. The default extension name for this file is "VAL".

6. **DATA POINT FILE.** The data point file is a sequential file which POLSPEC writes the keyboard entered data points for surface fitting. This file can also be created by any editor that creates an ASCII file. The data points can be entered into the surface fitting module from this file. This file is also needed to print the Hit Table of actual to predicted payoffs. The default extension for this file is "PNT".

POLSPEC file names may be up to twelve characters in length and take the following form:

Primary name (1-8 characters)
Period connector (".")
Extension (1-3 characters)

A typical file name might be "SELECT.DAT". This file name has a primary name "SELECT" and an extension "DAT".

POLSPEC has pre-defined extensions for all of the files it uses. A user may specify any DOS compatible extension for the files but it is often easier to use the defaults listed. A default extension can be specified by simply typing the "RETURN" key.

POLSPEC FILE	DEFAULT
Variable Name File	NAM
Policy Function File	POL
Policy Work File	WRK
Function Hardcopy File	PRN
Graphic Data File	VAL
Data Points File	PNT

The POLSPEC files listed above will all reside on the same disk drive. The default drive for these files is the "B" drive.

### III. USING THE PROGRAM

The system is set up by turning on all the components which will be used (computer, monitor, printer, and hard disk). If a hard disk is not being used, the POLSPEC startup diskette should be put into drive "A". Enter "POLSPEC", the program will then tell the user to insert the runtime diskette into drive "A". The policy file storage floppy should now be put into drive "B". The user is now ready to run the program.

POLSPEC will now prompt the user for one of the four program options:

"W"	is the policy work option. It allows the user to create a new policy, modify an existing one, view policy functions, output policy functions, etc.
"P"	is the policy output option. This option enables the user to output an entire policy or specific functions of choice. Functions can be output to the printer or a file of choice.
"R"	is the recovery option. If the program should terminate due to error or if there is a system failure, the user can recover any functions which were completed during the aborted session by using this program option.
"S"	is the surface fitting option. It allows the user to create a regression equation using data points and payoffs. The resulting equation is a fitted function, and a hit table of actual and predicted payoffs.

## Work Option

### File Specification

POLSPEC will initially prompt for the variable name and policy function files for the policy work session followed by the prompt for the label of the disk drive which will contain the name and policy files. The default drive is the "B" drive which can be indicated by typing the "RETURN" key. These prompts are shown below.

```
VARIABLE NAME FILE: _____ .NAM
POLICY FUNCTION FILE: _____ .POL
NAME, WORK, & POLICY FILE
DISK DRIVE: _
```

If the variable name file already exists, the names in the file will be picked up for use during the policy session. If the variable name file does not exist it will be created. The user then will be given an opportunity to enter variable names in numerical sequence starting with variable number one. The prompt session for initial variable names is illustrated below.

```
Enter the name
for variable
```

```
X 1: _____
```

Enter, in sequence, any or all names for the variables in the policy. When finished, type the "End" key.

Upon receiving the policy function file name and the work disk drive, POLSPEC will initially check if the policy function file already exists. If the file does not exist on the designated drive, POLSPEC will create a new policy work file with the same primary name as that given for the policy function file. For example, if the new policy function file name was entered as "EXAMPLE.POL" and if the disk drive is "B", then POLSPEC would create a work file named "EXAMPLE.TMP" on the "B" drive.

If the policy function file specified already exists, POLSPEC will first check to see if a work file with the same primary name also exists on the work drive. If the corresponding work file is already there, POLSPEC will leave it untouched. If the work file is not there, POLSPEC will create it, giving it the same primary name as the policy function file. POLSPEC will also transfer all of the function information from the policy function file to the policy work file.

During a policy specifying session, all of the policy work involves the policy work file (external storage) and computer memory (internal storage). The policy function file is only accessed at the beginning and end of a policy session. See Appendix A-1 for a detailed explanation of POLSPEC data processing procedures and file structures.

### Transactions

The following are the basic work option transactions:

LIST THE CURRENT FUNCTIONS	= L
DELETE A FUNCTION	= D
EXAMINE A FUNCTION	= E
GO TO FUNCTION KEY MODE	= F
TERMINATE THIS SESSION	= T

Work option transaction "L" lists the labels, function forms, and names of all functions currently resident in the work file. A sample listing is shown below.

F 1=F(X 1,X 2) MODEL 1	VARIABLE 1 - VARIABLE 2
F 2=F(X 3,X 4) MODEL 1	VARIABLE 3 - VARIABLE 4
F 3=F(F 1,F 2) MODEL 1	FUNCTION 1 - FUNCTION 2

Work option transaction "D" enters the user into a prompt session which enables the user to delete selected policy functions from the work file. This session is illustrated below.

Please enter the number of the policy function that you would like to delete.
--

>4

In this example, the user has typed in "4" which indicates that the function number 4 should be deleted from the work file. POLSPEC continues to prompt for functions to be deleted until the "End" key is entered.

Work option transaction "E" enables the user to examine a policy function within the scope of a fixed prompt cycle, as described above. This cycle consists of a series of prompt and display sessions, as illustrated below.

POLSPEC prompts the user for the numeric label of the function that the user wishes to examine.

Please enter the number of the policy function to be used.  
(ie. "1" for function number 1 )

> \_

If the user requests a function that already exists (resides in the work file) then POLSPEC will display the current function parameters on a special screen. The user will then be able to modify parameter values by choice. If a new function (not currently in the work file) is requested, POLSPEC will display "dummy" parameters on the special screen. These default values serve merely as sample values to aid the user in entering actual parameter values for the new function.

Sample function modification and function creation screens are shown in Tables A-2 and A-3. Note the special function key entries which are available with these screens. These keys provide a number of convenient shortcuts to the editing process.

The user will need to examine each function parameter as it is listed on the function modification or function creation screen. The user may keep a value shown on the screen by simply typing the "RETURN" key. When the user is done with the entries, type the "End" key. Then the user will be prompted for a final OK with respect to the set of parameter values displayed on the screen. If the set of parameters are OK, type in "Y"; if not, type in "N". "Y" causes the new set of parameter values to be written to the policy work file, replacing any old values that may have been in there for that function. "N" will cause the "cursor" to go back to the top of the screen and the session will be conducted again using the original values.

Table A-2. Sample Policy Function Modification Screen

FUNCTION NAME: VARIABLE 1 - VARIABLE 2 MATCH

F01 = F(X01,X02) Model 2

Y(ALO,BLO) 15    Y(ALO,BHI) -250  
Y(AHI,BLO) 35    Y(AHI,BHI) 100    YLO 0    YHI 100

ALO 40    AHI 95    BLO 40    BHI 100

STARTA 95    STOPA 40    INCA -5    STARTB 40    STOPA 100    INCB 5

AEXP 1    BEXP 3    ZERO SLOPE (A) 2    ZERO SLOPE (B) 1

KEEP THESE VALUES? Y

F1 <HELP>    RETURN <KEEP CURRENT VALUE>    End    <DONE>  
Esc    <CANCEL>

---

Table A-3. Sample Policy Function Creation Screen:

FUNCTION NAME: FUNCTION 1

F01 = F(X01,X02) Model 1

Y(ALO,BLO) 0    Y(ALO,BHI) 40  
Y(AHI,BLO) 60    Y(AHI,BHI) 100    YLO 0    YHI 100

ALO 0    AHI 100    BLO 0    BHI 100

STARTA 100    STOPA 0    INCA -10    STARTB 0    STOPA 100    INCB 10

AEXP 1    BEXP 3    ZERO SLOPE (A) 2    ZERO SLOPE (B) 1

KEEP THESE VALUES? \_

F1 <HELP>    RETURN <KEEP CURRENT VALUE>    End    <DONE>  
Esc    <CANCEL>

---

A policy function is a mathematical function of two attributes. We shall label these attributes "A" and "B" for convenience in referring to them. A function attribute may be either a base variable (lowest level of the function hierarchy) or a function defined at a lower level of the hierarchy. The attributes of policy function may be non-interactive or interactive. A function may be linear or non-linear with respect to either attribute.

The parameters for a given policy function consists of 27 separate items. Some of these are simply names or labels, such as the function name and attribute designators. Other parameters are numeric values to be used in mathematical calculations. The function parameters are listed and described below. Note that some parameters can take on either integer or real values, while others can only be represented as integers. The parameter listing is annotated to indicate what type of parameter input is expected.

#### PARAMETER LIST:

Function Name	The function name describes in a few words the relationship between the two function attributes. A function pairing a student's grade point average and SAT score, for example, might be "Academic Potential". A function name may contain up to 44 characters.
Function Number (integer only)	This is simply the function's numeric label (ie. "1").
Attribute Designator	Either "X" or "F"; "X" for a variable, "F" for a function
Attribute Number  (integer only)	Numeric label for an attribute. (ie. "1")

The function number and attribute designators and numbers are combined into a "Function Form" such as:

$$F01 = F(X01,X02)$$

**Model Number**  
(integer only)

There are two available base models which can be used in specifying a policy function. Model 1 allows for a single zero-slope point for the function, with respect to either attribute. Model 2 allows two zero-slope points with respect to the second attribute only.

**Policy Function  
Payoff  
Cornerpoints**  
(real/integer)

The function payoff cornerpoints are those values ("Y"s) which correspond to the four combinations of function attribute extreme values. The extreme values are labelled LO (low) and HI (high) so we have:

Y(ALO,BLO)    Y(ALO,BHI)  
Y(AHI,BLO)    Y(AHI,BHI)

**Payoff Range  
Limits**  
(real/integer)  
(real/integer)

These limits refer to the desired payoff value range for displayed or printed payoffs. The low limit (YLO) and the high limit (YHI) may differ from the Y(ALO,BLO) and Y(AHI,BHI) values discussed above. The cornerpoints are crucial to the formulation of the policy function itself. The payoff range limits have nothing to do with the function equation. They merely set a boundary on what payoff values will be printed in the table. A typical use is to convert negative payoff values to zero.

**Low and High  
Values for  
function  
attributes**  
(real/integer)

The low and high values for the "A" attribute are labeled ALO and AHI respectively. BLO and BHI are the low and high values for the "B" attribute.

**Table Reference  
Values of  
Attributes**  
(integer only)

These values determine which "A" and "B" pairings will be used to calculate function payoff values for both the payoff table and the sorted payoffs listing.

STARTA = the starting "A" reference value  
STOPA = the ending "A" reference value  
INCA = the amount by which STARTA is  
incremented to arrive at STOPA

STARTB, STOPB, and INCB are the reference values for the "B" attribute.

INCA and INCB can be expressed as either negative or positive values depending on whether the starting value is greater or less than the stopping value. In either case, POLSPEC checks the STARTA, STOPA, STARTB, and STOPB values in order to determine whether INCA and INCB should be positive or negative.

Function  
Attribute  
Exponents  
(real/integer)

Function curvilinearity can be specified with respect to either attribute by adjusting the exponents for the "A" and "B" attributes. Exponents of 1 for both attributes will result in a function that is linear with respect to both attributes.

AEXP = exponent for "A" attribute

BEXP = exponent for "B" attribute

Zero Slope  
Points  
(integer only)

If the function is curvilinear with respect to one or both attributes, the user can specify where the function "flattens out" or has a zero slope with respect to an attribute. This zero slope point can occur at either the low or high end of the attribute value range. If the function is linear with respect to an attribute, the zero slope parameter for that attribute has no effect.

Where: ZERO SLOPE (A) = zero slope point for the function with respect to the "A" attribute.

ZERO SLOPE (B) = ..."B" attribute.

IF: Zero SLOPE = 1 Point function "flattens out" at low end with respect to the attribute

Zero SLOPE = 2 Point function "flattens out" at high end with respect to the attribute

In the sample policy function featured in Table A-2, The "B" attribute had an exponent of "3" and a zero slope point of "1". It can be seen that the function "flattens out" with respect to the low end of the "B" attribute. The function is linear with respect to the "A" attribute, so the zero slope point value for "A" doesn't matter.

When a function form (i.e.  $F01 = F[X01,X02]$ ) is specified POLSPEC will check whether variable attributes already have names. POLSPEC will prompt the user for a name if a designated attribute does not yet have one.

The user will be prompted for function display viewing options, as shown below, when finished with examining the parameters of the chosen function.

View the policy function TABLE	Y
View the function PARAMETERS	Y
View the function EQUATION	Y
View a table of SORTED PAYOFFS	Y
Use REAL FORMAT for payoff values	N

In this example, the user has chosen to view the function table, parameters, equation, and sorted payoffs. The table payoffs would be displayed as integer values.

A sample prompt for the sample function featured in a previous example, is shown below. The contents of the sample output correspond to the viewing options provided above.

After viewing the payoff table the user is in a position to decide whether or not to "save" any changes made for an existing function, or save the entire set of parameter values that were specified in the case of a new function. POLSPEC will now prompt the user for the "save" decision as shown below.

Do you wish to save this function?

> \_

Table A-4. Policy Function Output

Header, Parameters, Equation, and Payoff Table

\*\*\*\*\* POLICY SPECIFYING SYSTEM \*\*\*\*\*

POLICY: C:USER.POL NAMES: C:USER.NAM

F01=F(X01,X02) MODEL 2 VARIABLE 1 - VARIABLE 2 MATCH  
X01=VARIABLE 1 X02=VARIABLE 2

Y(ALO,BLO)= 15.0 Y(ALO,BHI)=-250.0 Y(AHI,BLO)= 35.0 Y(AHI,BHI)= 100.0  
AEXP= 1.0 BEXP= 3.0 ZERO SLOPE (A)=1 ZERO SLOPE (B)=1  
RLO= .0 RHI= 100.0 ALO= 40.0 AHI= 95.0 BLO= 40.0 BHI= 100.0  
STARTA= 95 STOPA= 40 INCA= -5 STARTB= 40 STOPB= 100 INCB= 5

$$\begin{aligned}
 F01 = & 35.00 + .3636E+00 * (X01 - 95.00) ** 1.0 \\
 & + .5417E-01 * (X02 - 40.00) ** 2.0 \\
 & + .1136E-04 * (X01 - 95.00) ** 1.0 * (X02 - 40.00) ** 3.0 \\
 & + .9848E-03 * (X01 - 95.00) ** 1.0 * (X02 - 40.00) ** 2.0 \\
 & - .6019E-03 * (X02 - 40.00) ** 3.0
 \end{aligned}$$

F01	X02													
	40	45	50	55	60	65	70	75	80	86	90	95	100	
95	35	36	40	45	52	59	68	76	83	90	95	99	100	
90	33	34	37	42	48	54	60	65	70	73	74	73	68	
85	31	32	35	39	43	48	52	55	56	56	53	46	36	
80	30	30	33	36	39	42	44	45	43	39	31	20	5	
75	28	28	30	33	35	36	36	34	30	22	10	0	0	
X01 70	26	27	28	30	31	31	29	24	16	5	0	0	0	
65	24	25	26	26	26	25	21	14	3	0	0	0	0	
60	22	23	23	23	22	19	13	4	0	0	0	0	0	
55	20	21	21	20	18	13	5	0	0	0	0	0	0	
50	19	19	19	17	14	7	0	0	0	0	0	0	0	
45	17	17	16	14	9	2	0	0	0	0	0	0	0	
40	15	15	14	11	5	0	0	0	0	0	0	0	0	

Table A-5. Policy Function Output: Listing of Sorted Payoffs

\*\*\*\*\* POLICY SPECIFYING SYSTEM\*\*\*\*\*

POLICY: C:USER.POL  
 NAMES: C:USER.NAM

F01=F(X01,X02) MODEL 2 VARIABLE 1 - VARIABLE 2 MATCH  
 X01=VARIABLE 1  
 X02=VARIABLE 2

\*\*\*\*\* SORTED PAYOFF VALUES \*\*\*\*\*

F01/PAYOFF	100	99	95	90	83	76	74	73	73	70
X01/INDEXA	95	95	95	95	95	95	90	90	90	90
X02/INXEXB	100	95	90	85	80	75	90	85	95	80
F01/PAYOFF	68	68	65	60	59	56	56	55	54	53
X01/INDEXA	95	90	90	90	95	85	85	85	90	85
X02/INXEXB	70	100	75	70	65	80	85	75	65	90
F01/PAYOFF	52	52	48	48	46	45	45	44	43	43
X01/INDEXA	95	85	90	85	85	95	80	80	85	80
X02/INXEXB	60	70	60	65	90	55	75	70	60	80
F01/PAYOFF	42	42	40	39	39	39	37	36	36	36
X01/INDEXA	90	80	90	85	80	80	90	95	85	80
X02/INXEXB	55	65	50	55	60	85	50	45	100	55
F01/PAYOFF	36	36	35	35	35	34	34	33	33	33
X01/INDEXA	75	75	95	85	75	90	75	90	80	75
X02/INXEXB	65	70	40	50	60	45	75	40	50	55
F01/PAYOFF	32	31	31	31	31	30	30	30	30	30
X01/INDEXA	85	85	80	70	70	80	80	75	75	70
X02/INXEXB	45	40	90	60	65	40	45	50	80	55
F01/PAYOFF	29	28	28	28	27	26	26	26	26	25

Table A-5. Concluded

X01/INDEXA	70	75	75	70	70	70	65	65	65	65
X02/INXEXB	70	40	45	50	45	40	50	55	60	45
F01/PAYOFF	25	24	24	23	23	23	22	22	22	21
X01/INDEXA	65	70	65	60	60	60	75	60	60	65
X02/INXEXB	65	75	40	45	50	55	85	40	60	70
F01/PAYOFF	21	21	20	20	20	19	18	16	14	13
X01/INDEXA	55	55	80	55	55	60	55	70	65	60
X02/INXEXB	45	50	95	40	55	65	60	80	75	70
F01/PAYOFF	13	10	5	5	5	4	3	0	0	0
X01/INDEXA	55	75	80	70	55	60	65	75	75	70
X02/INXEXB	65	90	100	85	70	75	80	95	100	90
F01/PAYOFF	0	0	0	0	0	0	0	0	0	0
X01/INDEXA	70	70	65	65	65	65	60	60	60	60
X02/INXEXB	95	100	85	90	95	100	80	85	90	95
F01/PAYOFF	0	0	0	0	0	0	0			
X01/INDEXA	60	55	55	55	55	55	55			
X02/INXEXB	100	75	80	85	90	95	100			

If the answer is "Y", any parameter entries just made will be saved. If the user made no changes to an existing function, the old parameters will remain unchanged.

If the answer "N", any parameter entries just made will not be saved. If the function was a new one, it will be effectively erasing the function. If the function was an existing one, the parameter values in effect at the start of the modification session, will be retained instead.

Whether or not the user "saved" parameter entries made; the user has the option of having a copy of the policy function output sent to the printer or to a file of user designation. The POLSPEC prompt for a "hardcopy" as shown below.

```
Would you like a hardcopy of this function?  
> _
```

If the user answers "N", no hardcopy will be produced.

If the user answers "Y", a prompt for whether the hardcopy should be sent to the printer or to a file will be displayed. If the hardcopy is sent to the printer, the user will also be prompted for the number of copies to be printed. If a file is selected, the user will be prompted for the hardcopy file name. Whether the printer or a file is chosen for the hardcopy, the user will be prompted for the same output options that were offered for viewing the policy function (table, parameters, equation, sorted payoffs, real format). In addition, if the printer is chosen, the width of the paper can be specified (14"). The default is narrow (11") paper.

The final step in the "fixed cycle" is the prompt session for the graphic data option. If the user wishes, function attribute and payoff information can be written off to a special "graphic data" file for use in plotting the policy function with a separate graphics package. POLSPEC prompts as follows:

```
Do you wish to save the "A" attribute, "B" attribute,  
and function payoff values to a file for future  
graphing?  
> _
```

If the user answers "Y", POLSPEC performs the graphic data transfer. To produce graphics many off-the-shelf products are available.

Function Key Mode is a means whereby an experienced POLSPEC user can do policy work in a less restricted manner than in the fixed cycle described previously. To enter the Function Key Mode, type in "F" as the work option transaction. The user will enter into Function Key Mode as indicated by the following prompt screen. The function key entries in Function Key Mode are the only entries which do not require the use of the RETURN key. A function key keystroke suffices.

**FUNCTION KEY MODE:**

F3 <LIST>  
F4 <DELETE>  
F5 <WORK>  
F6 <VIEW>  
F7 <OUTPUT>  
F8 <VALUES>

F1 <HELP>    F10 <DONE>

All of the features of the fixed cycle are offered on this screen.

- F3 --> Lists the current work file functions.
- F4 --> Brings up the function deletion prompt session.
- F5 --> Prompts for a function number and then brings up either the function modification or function creation prompt session depending on whether or not the function already exists. This will not allow you to create surface fitting functions.
- F6 --> Prompts for a function number and then brings up the viewing option prompt and display session.
- F7 --> Prompts for a function number, output device (printer or file), and output options, just as in the "hardcopy" feature of the fixed cycle, except that only one function can be printed out at a time.
- F8 --> Prompts for a function number and output device just as in the graphic data feature of the fixed cycle.
- F10--> Terminates Function Key Mode and returns you to the work mode option menu.

After any function key operation (F3 through F8) POLSPEC will return to the Function Key Mode menu. When using function key operations F6, F7, and F8, POLSPEC will check first for the existence of the function of choice. If the function does not exist, POLSPEC will display a message to that effect and return to the function key mode menu.

The work mode option "T" requests the termination of the current policy specifying session. When the user is done with the policy work type in "T" to exit. Upon entering "T", POLSPEC will prompt for the name of a policy file in which to permanently store the policy functions (which currently reside in the work file). This prompt is shown below.

Enter the name of the file which will store your policy.

POLICY FILE: \_\_\_\_\_ .POL

F1 <HELP>    RETURN <overwrite old file>    End <no save>

At this point refer to the policy file designated at the beginning of the policy session by typing the "RETURN" key. The current functions will then be transferred from the work file to the policy file, overwriting any functions that may be in there.

The user may also enter a policy file name other than the one given at the start of the session. This would be appropriate in the case where the user wants to build an experimental policy based on an old policy, which the user wishes to retain.

If the user doesn't want to save the policy functions worked on, then type in the "End" key. The functions will remain in the work file without being transferred to a policy function file.

### Policy Function Output Option

POLSPEC offers a policy function output option ("P") which enables the user to send functions of user choice to either the printer or to a file. This option assumes there is an existing policy from which function output can be derived. The user cannot do any other function work under this option.

### File Specification

Upon designation of the "P" option, POLSPEC will prompt the user for the variable name file, function policy file, and work disk drive, just as with the work option previously

discussed. Upon receipt of this information, POLSPEC will check for the presence of the variable name, policy work, and policy function files. If one or more of these are not on the work drive, POLSPEC will proceed as follows.

NO variable name file:	POLSPEC will prompt you for a set of variable names as it would for a policy work session.
NO policy work file:	POLSPEC will set up a work file for this session.
NO policy function file:	"P" option cannot be performed without an existing policy file. POLSPEC will display a message to that effect and then terminate the session.

#### Designation of Output Device

At this point, POLSPEC will prompt for the output device. The user may have the function output go to the printer or to a file. If the user chose to have the output go to the printer, the user will be prompted for the number of copies of the set of functions to print. For example, if the user chose to get function output for function numbers 1,2, and 3, and instruct POLSPEC that five copies are desired; POLSPEC will provide five sets of the three functions. POLSPEC prints the three functions in secession, and then loops back to repeat the operation as many times as is necessary to provide the correct number of function output sets.

If the user chose to have the output go to a file the user will be prompted for the name of the function output file.

#### Designation of Policy Functions

POLSPEC will prompt the user for the set of policy functions to be used in this session as shown.

Please enter the numbers of the policy functions that you would like output, one entry at a time.

FUNCTION NUMBER: \_\_

F1 <HELP> RETURN <output all functions> End <Done>

In this prompt session the user can designate specific functions for output by typing a function number followed by the "RETURN" key. POLSPEC will signal its acceptance of the entry by erasing it and standing by for the next one. When the user is done entering in function numbers the user types the "End" key to end this prompt session. The user may have all of the functions of the policy set up for output by simply entering the "RETURN" key as the only input.

#### Policy Function Output Print Options

When the set of policy functions to be output has been defined, POLSPEC will prompt the user for the print options to be used for this output session.

Print the policy function TABLE	Y
Print the function PARAMETERS	Y
Print the function EQUATION	Y
Print a table of SORTED PAYOFFS	Y
Use REAL FORMAT for payoff values	N
Set up for WIDE PAPER (14")	N

These are the same print options offered under the POLSPEC work option for policy function hardcopies. Under the POLSPEC output option, these print settings will apply to the entire set of functions to be output. The user cannot have one set of print options applied to one function and a different set of print options applied to another function.

#### Designation of Data Points for Surface Fitting Functions

Whenever a surface fitting function is to be output the user has the option of printing the hit table. To print this table a file of data points must be specified. The user will be prompted for a data point file for each surface fitting function. If the user doesn't require the hit table, press "End" or "Esc" to kill the rest of the print run. If a file is specified, the user will also be prompted for the criterion data point.

### Recovery Option

There may be times while running POLSPEC that something happens to "bomb" the program in the middle of a policy work session. If this should happen, the user will not have lost the policy functions that were complete at that point in time.

The policy work file contains the current versions of policy functions. This file will not be compromised by an error termination of POLSPEC or even by a system crash (caused by power surge, failure, etc.). For this reason, the policy work file will be the basis for recovery from a "bombed out" program or a system crash.

To recover, run POLSPEC with the "R" option. POLSPEC will prompt the user for the name of the policy work file used during the aborted session. The policy work file name is the same as the name that is used for the policy function file during the aborted session, except that the extension is "TMP". For example, if during the aborted policy session, the policy function file was "MYJOB.POL", then the policy work file for that session was "MYJOB.TMP". This is the name POLSPEC now needs to do the recovery. POLSPEC also needs the name of the policy file to which the recovered policy functions should be transferred. The user may use the same policy function file that was used for the aborted session, or may designate a new file.

POLSPEC will transfer all completed and undamaged policy functions from the policy work file to the policy function file.

### Surface Fitting Option

This module uses a stepwise regression algorithm to calculate the regression weights for a surface fitting equation. The user will be prompted for data points and payoff values. The equation generated is of the following form:

$$Y = A0 + [ A1 * (X1**EXP1) * (X2**EXP2) ] + [ A2 * (X1**EXP1) * (X2**EXP2) ] + \dots + [ A19 * (X1**EXP1) * (X2**EXP2) ]$$

The user may specify a function having up to 19 terms. The user will be prompted for the exponents for each of these terms. The exponent values may be any real number with in the range of 0 to 6.0.

1. This module starts out by prompting for the variable name and policy files.
2. POLSPEC prompts for the numeric label of the function that the user wishes to work with.

Please enter the number of the policy function to be used.  
(i.e. "1" for function number 1)

> \_

3. If the user requested a function that doesn't exist, the user will be asked if a new function should be created. The user would respond by entering either a "Y" for Yes or "N" for No. If the response is yes continue to step 4. If the response is a no the user will be asked if another function is requested. A yes response would return the user to step 2, else the user will be asked to save the policies developed and then return to the main menu.

If the user requests a function that already exists, the user will be able to either overwrite the function with a surface fitting function, view the function, or print out a hard copy of the function.

4. The user will specify the function name and a function form. POLSPEC will check whether variable attributes already have names. POLSPEC will prompt for a name if a designated attribute doesn't yet have one.

Function Name: \_\_\_\_\_

F 1 = F(X\_\_ , X\_\_) MODEL 0

The surface fitting functions can only contain variable attributes of the lowest level. In other words, unlike policy functions, the user can't have a function containing another function (i.e.  $F3 = F(F1, F2)$ ).

Surface fitting functions are all designated as model type 0 functions.

5. The user will now be presented with two options for specifying the functional form of the generated equation. Option 1 allows the user to specify the number of terms the equation will contain, and the exponent values of the equation. Option 2 will use POLSPEC default values.

6. If the user chooses option 1 in step 5, the user will have to enter the exponent data. This is done on the upper part of the next screen. If option 2 was specified this portion of the screen is skipped and the user will continue with the lower portion of the screen. The upper portion of the screen follows:

```
TERMS IN EQUATION --> _  
PROMPTING FOR TERM  
X1 EXPONENT --> ____  
X2 EXPONENT --> ____
```

The user may specify up to 19 terms. The prompt screen will tell the user what term it is prompting the exponents for. The exponent values may be any real number between 0 and 6.0.

After all the terms have been entered, the equation will be displayed. Verify the equation and respond with a "Y" if the equation is correct. A "N" response will allow the user to re-enter the values starting with the number of terms in the equation.

The lower portion of the screen is used for printing out the payoff table. The values entered here will determine the parameter pairings used in the calculating the payoff table.

```
X1 TABLE PARAMETERS:  START  100  STOP  0  INCREMENT  -10  
X2 TABLE PARAMETERS:  START   0  STOP 100  INCREMENT  10  
F2 PAYOFF LIMITS:      MIN      0  MAX  100  
  
KEEP THESE VALUES _
```

7. The user will now be presented with two options for entering the data points and payoff values. Option 1 allows POLSPEC to read the data from a file. Option 2 will allow the user to enter data from the keyboard.

Option 1 will display the following prompt:

```
Enter file name _ : _____ .PNT
Of the 3 data points, which is the criterion (1,2 or 3)--> _
```

Enter the name of the file containing the data points. The next prompt asks for the criterion data point. In other words, which of the points is the payoff value. The user would respond with either a 1, 2 or 3. Which ever value is chosen for the criterion, the other two points will be the X1 and X2 data points used in the generated equation.

If option 2 was chosen, POLSPEC will display the following prompt:

```
Save Data Points to _ : _____ .PNT
Of the 3 data points, which is the criterion (1,2 or 3)--> _
Data Points --> _____ , _____ , _____
```

Enter the name of the file to receive the data points. This file is needed later to produce the hit table of actual to predicted payoffs. The user also needs to specify which of the three points is the criterion or payoff point.

The data points must be integer values. After entering one value, press return. A "," will appear after the last value. The user may then enter the next value. The user will not be able to enter any integer more than 5 digits long.

The user must be careful when entering many data points that contain large numbers. Due to the algorithm used, some very large numbers are calculated internally. It is possible to cause an overflow condition if the user specified large exponents when defining the equation. If for some reason this becomes a problem, either use fewer data points or scale the values down.

8. The user now has the option to print some summary statistics from each step of the stepwise algorithm. This is the only time the user will be able to view these values. They are intermediate values computed in the algorithm itself. They can't be looked at later when the function is viewed or printed out. The results of the last step will always be displayed. The output looks as follows:

STEP 1, MULTIPLE RSQ = .6923, NUMBER OF PREDICTORS = 1  
STD.DEV. FOR RESIDUALS = 2.8284E+01

ANALYSIS OF VARIANCE

	DF	SUM SQUARES	MEAN SQUARE	F RATIO	P LEVEL
REGRESSION	1		3.60000E+03	3.60000E+03	
4.50000E+00	.1679				
RESIDUAL	2	1.60000E+03	8.00000E+02		

STEP 2, MULTIPLE RSQ = 1.0000, NUMBER OF PREDICTORS = 2  
STD.DEV. FOR RESIDUALS = .0000E+00

ANALYSIS OF VARIANCE

	DF	SUM SQUARES	MEAN SQUARE	F RATIO	P LEVEL
REGRESSION	2		5.20000E+03	2.60000E+03	
2.60000E+03	.0139				

The user is now prompted to direct the output to either the screen, printer or a file.

Direct output to Screen, Printer or File (S,P or F) --> \_

If the user responds with a "S" the Summary Statistics will be directed to the screen. If the user specifies a "P" the printer will receive the output. A "F" response will direct output to a file.

If output is to be directed to a file they following prompt will appear.

FUNCTION OUTPUT FILE: \_\_\_\_\_ .PRN

The file specified will receive the output.

9. POLSPEC now prompts the user for the viewing options. Enter a "Y" for desired products. Press END when the user has completed selecting the options. The products specified will then be displayed.

10. After viewing the policy the user is asked whether to to save the policy function.

Do you wish to save this policy function?

>\_

Respond with a "Y" if the policy should be saved to the work file. A "N" response will cause the policy not to be saved.

### ERROR HANDLING

POLSPEC has facilities for error checking with respect to the inputs it receives. Most often, an error will involve an illegal character or an out of bounds numeric entry. Occasionally an error will result upon the use of fractional exponents for the policy function parameters, AEXP, BEXP and the surface fitting exponent parameters.

If an error occurs, POLSPEC will interrupt the current prompt session in order to issue the appropriate error message. The user will be able to return to the prompt session after reading this message, by typing the "RETURN" key. Any entries made on the prompt screen, prior to the error interrupt, will be restored, and the cursor will be positioned at that item on which the error occurred. The user will now be able to reenter the input and continue on with the prompt session.

**APPENDIX B**

**POLICY CAPTURING PROGRAM**  
**(POLCAP)**

**USER'S MANUAL**

**RELEASE 1.0**

## TABLE OF CONTENTS

	Page
I. INTRODUCTION .....	41
II. GENERAL PRINCIPLES AND FEATURES .....	43
III. USING THE PROGRAM .....	44
Create a New Decision Context .....	44
Revise an Existing Context .....	46
Delete an Existing Context .....	48
Generate Sample Data .....	48
Select Attribute Sets/Judges .....	48
Delete Attribute Set .....	50
Collect Judgments .....	50
Compute Interrater Reliability .....	50
Analyze Judgments .....	51

## LIST OF FIGURES

Figure	Page
B-1 Functions of POLCAP Options .....	45

## LIST OF TABLES

Table	Page
B-1 Judgment Profile for Policy Capturing in Promotion Application .....	42
B-2 Summary Information on Attributes .....	46
B-3 Attribute Changes Within a Context Revision .....	46
B-4 Changes to Attribute Correlation Matrix .....	47
B-5 Selection of Attribute Sets for Judging .....	49
B-6 Assigned Rating for Case 1 .....	50
B-7 Interrater Reliability Analysis .....	51
B-8 Regression Results .....	52

## I. INTRODUCTION

Policy capturing is a technique that attempts to replicate observed decisions by developing a mathematical function which combines decision attributes. In order to apply the policy capturing technique to the construction of a decision model, the analyst must first develop a representative decision set which is presented to a judge or judges (decision makers) holistically. Comprising the decision set are a number of profiles, each of which describe an option in the decision process. Each profile is complete with those attributes thought to be most relevant to the decision. Attributes are described in terms of an exact value for each continuously measured attribute and a categorical value for discrete attributes. The set of profiles should be selected such that a variety of combinations of attributes are represented. The decision maker is then asked to provide a judgment as to the overall value of each profile, either in naturally occurring units (dollars, for example), some arbitrary scale (for example, 0 to 100), or a rank order within the decision set. To complete the process of policy capturing, linear regression is used to regress the attributes, using the rank order or score as the dependent variable and the attributes as the independent or predictor variables. The resulting equation represents the "captured policy." In the case of multiple judges, a clustering technique is used to determine the single equation that best represents the group decision process.

Selection of the profiles to be judged is one of the keys to proper utilization of the technique and should take several factors into account. First, the set of profiles should be comprehensive; that is it should reflect the range of values for each attribute. Second, no single area of the scale for a continuous attribute, nor single category for discrete variables, should be allowed to dominate the set; it should be balanced across the range of values for each of the attributes. Third, the set of profiles should be reflective of the real world. Unlikely or impossible combinations of attributes should occur infrequently or not at all. The number of attributes is closely related to the decision maker's ability to process information. Generally, seven attributes is the maximum number that should be considered in applications of policy capturing. There is no definitive guidance as to the number of profiles that need to be included in the decision set. Since regression is the method of analysis, the rule of thumb may apply that there should be a minimum of 10 cases (profiles) per attribute for which a weight parameter is to be estimated. There may also need to be additional cases when the stability of the parameter estimates is in question such as would be caused by a strong multicollinearity problem among the attributes.

A brief description of how the policy capturing technique is utilized may be observed in the following example of the Weighted Airman Promotion System (WAPS). This is a system used by the Air Force to determine which individuals eligible for promotion will in fact be selected. This section uses a number of personnel attributes to determine a rating for each candidate. In this example six attributes were chosen. They are: scores determined from a skill knowledge test (SKT) and the promotion fitness examination (PFE), time in service (TIS), time in grade (TIG), awards and decorations (AD), and an individual

performance rating (IPR). The SKT and PFE would be tests of the airman's knowledge of a specific job specialty and of general military subjects and management practices at the appropriate level. In this promotion system TIS and TIG will be measured in months. The scores for awards and decorations is assigned according to their order of precedence. For example, combat related decorations receive higher value and scores than non-combat service awards. The IPR score is calculated by averaging the most recent enlisted performance reports, given annually by the airman's supervisor, and then multiplying the average by a weighting factor.

In the context of the promotion example, a set of profiles, each representing a candidate for promotion, would be presented to the decision maker (or panel of decision makers) in a form similar to that shown in Table B-1. The decision maker would then be asked to respond with a score for each individual, representing the individual's relative "promotability." The score can be expressed on some arbitrary scale (for example, 0-100), in terms of the point scales used to measure each attribute, a rank ordering, or even categories (1-5 categories receiving 1 to 5 points each).

<b>Table B-1. Judgment Profile for Policy Capturing in Promotion Application</b>						
	<b>Skill Knowledge Test Score</b>	<b>Promotion Fitness Examination Score</b>	<b>Time In Service</b>	<b>Time In Grade</b>	<b>Awards &amp; Decorations</b>	<b>Individual Performance Ratings</b>
<b>Applicant</b>						
1	86	62	47	18	22	135
2	77	81	106	8	18	133

The profiles would have been chosen to ensure that the entire set is both representative of the pool of applicants who would normally come before a promotion board and that it includes adequate coverage of the range of each attribute. The profile set would also be inspected (by examining the intercorrelations between attributes in the set) for a lack of independence between attributes. In addition, the set would be screened to eliminate unlikely or impossible applicants (for example, an applicant who has the Congressional Medal of Honor and three Purple Hearts, but only one month time-in-service). After the judges have completed their task, the value model is developed by performing a regression analysis on the judged profiles. In a group situation, the individual judge's equations are clustered using a mathematical clustering routine, in order to arrive at a single equation of promotability. The resulting clusters of equations are then examined to determine how many different rating patterns were evidenced by the rating panel. Aberrant raters can be removed so that the equations can be smoothed to a single equation, or feedback techniques can be used to eliminate group differences.

## II. GENERAL PRINCIPLES AND FEATURES

POLCAP is menu driven similar to POLSPEC, PAYGEN, etc. The edit facilities in POLCAP are not as robust as some of the other programs but the POLCAP algorithm is simpler and the program runs extremely fast.

Only one keyboard entry may be made at a time.

Default values can usually be entered by typing only the RETURN key.

Program prompt - User reply transactions are mediated through the use of special prompt screens. Normally, the first choice displayed on a prompt screen will be shaded; to select this choice just type the RETURN key. Other choices may be selected by typing their associated number. An exception to this requires the user when being prompted to select a decision context or an attribute set that are not listed in the first position, to move the shaded area to the desired position by means of a cursor key.

No more than 20 attributes can be entered for a context and only 20 contexts are allowed in this program. Each context must have at least 2 attributes otherwise it will not be allowed to enter.

Some of the prompt screens allow for other requests such as:

"RETURN" --> keep the current value or use the default value
"F2" --> change data drive
"END" --> end this prompt session - keep the changes
"ESC" --> end this prompt session - cancel this entry

The user should record for subsequent use in the program, the following inputs:

Attribute name
Context name
Judge's name and 3-digit identification number
Seed value for program generated data files

### III. USING THE PROGRAM

After viewing an introductory screen the user is presented with the following options:

- 1 - Create a new decision context.
- 2 - Revise an existing context.
- 3 - Delete an existing context.
- 4 - Generate sample data.
- 5 - Select attribute sets/judges.
- 6 - Delete attribute set.
- 7 - Collect judgments.
- 8 - Compute interrater reliability.
- 9 - Analyze judgments.

Figure B-1 displays the functions of POLCAP options.

#### Create a New Decision Context

When entering a new decision context, the user will be asked to respond to the following prompts:

Name of Attribute 1 -->  
Mean Value of Attribute -->  
Standard Deviation of Attribute -->  
Maximum Value of Attribute -->  
Minimum Value of Attribute -->

The user may continue to enter attributes by keying RETURN; after the required number of attributes have been entered the user keys in the number "0".

The program then requests a description (name) for this new context and displays in summary form previously inputted information for each attribute along with a request for correlations for all combinations of variable pairs, such as 1-2, 1-3, 2-3, etc. This information is shown in Table B-2.

<b>Options</b>	
<p>1 - Create a new decision context.</p> <p>2 - Revise an existing context.</p> <p>3 - Delete an existing context.</p>	<p>Creates/Revises/Deletes a context. Requires names for context and attributes, and Means, SD, MAX, and MIN. values for each attribute. Also requires correlations for each pair of attributes.</p>
<p>4 - Generate sample data.</p>	<p>Option generates required data for user requested sample size and number of judges.</p>
<p>5 - Select attribute sets/judges.</p>	<p>User selects from inputted data desired attribute set for a given context. User assigned by name and number judges to specific context and attribute sets.</p>
<p>6 - Delete attribute set.</p>	<p>User may delete any attribute set in any context.</p>
<p>7 - Collect judgments.</p>	<p>Requires the user to rate each case that has been generated in option 4. Simulates judges rating activity.</p>
<p>8 - Compute interrater reliability.</p>	<p>Develops correlation matrix for interrater reliability analysis.</p>
<p>9 - Analyze judgments.</p>	<p>Displays results of regression analysis in tabular form.</p>

**Figure B-1.** Functions of POLCAP Options

<b>Table B-2. Summary Information on Attributes</b>					
<b>Number</b>	<b>Attribute Name</b>	<b>Mean</b>	<b>S.D.</b>	<b>Max</b>	<b>Min</b>
1	SKT	88.00	2.40	100.00	52.00
2	PFE	76.00	3.10	90.00	65.00
3	TIS	45.00	2.20	65.00	23.00
Enter correlation for 1, 2 --> .88					

### Revise an Existing Context

This option provides the means to revise any attribute information previously inputted for an existing context including the correlations.

The program prompts the user to identify the context and then displays to the user information relative to attribute 1. After the changes have been made, the program then displays similar information for the remaining attributes, in turn, allowing for changes to those attributes. An example of this procedure is displayed in Table B-3.

<b>Table B-3. Attribute Changes Within a Context Revision</b>	
Name of Attributes -->	SKT
Mean Value of Attribute -->	66.00
Standard Deviation of Attribute -->	4.50
Maximum Value of Attribute -->	95.00
Minimum Value of Attribute -->	0.00
Press RETURN to change nothing, 1 for name, 2 for mean, 3 for standard deviation, 4 for maximum, 5 for minimum, 0 to stop viewing attributes.	

After the attribute changes have been completed, the user is then shown a correlation matrix for the attributes and is given the opportunity to input new values where needed. An example of this is shown in Table B-4.

**Table B-4. Changes to Attribute Correlation Matrix**

Before Change to R1, C2							
A		1	2	3	4	5	6
	1	1.00	0.80	0.75	0.70	0.05	0.05
	2	0.80	1.00	0.75	0.70	0.06	0.02
	3	0.75	0.75	1.00	0.85	0.07	0.06
	4	0.70	0.70	0.85	1.00	0.70	0.03
	5	0.05	0.06	0.07	0.70	1.00	0.50
	6	0.05	0.02	0.06	0.03	0.50	1.00
	Enter row number ( $\emptyset$ to end program) --> 1 Enter column number (cannot be same as row number --> 2 Enter value for this cell --> .85						
After Change to R1, C2							
B		1	2	3	4	5	6
	1	1.00	0.85	0.75	0.70	0.05	0.05
	2	0.85	1.00	0.75	0.70	0.06	0.02
	3	0.75	0.75	1.00	0.85	0.07	0.06
	4	0.70	0.70	0.85	1.00	0.70	0.03
	5	0.05	0.06	0.07	0.70	1.00	0.50
	6	0.05	0.02	0.06	0.03	0.50	1.00
	Enter row number ( $\emptyset$ to end program)-->						

### Delete an Existing Context

The program displays all existing contexts and prompts the user to select the context to be deleted. After the context has been selected, the program request an additional confirmation that the context is to be deleted. If an affirmative response is given, the context and all its associated information is then removed.

### Generate Sample Data

Under this option a sufficient number of profiles are generated to provide the desired number of cases for each judge. This option can also be used to replace existing data files with new data. The basic prompts and their restrictions are:

1. Choose context
2. Enter sample size (Max 100)
3. Enter number of judges (Max 10)
4. Enter seed value for this file

After the required inputs have been satisfied, the program then sums the statistics and computes means for all values.

### Select Attribute Sets/Judges

The first part of this option lets the user select from a given context any or all of the attribute sets previously inputted for that context. These are the sets that will be given to the judges. An example of this is displayed in Table B-5. Part A presents the user with the attributes and their statistics previously assigned to the selected context. Part B displays the results of the users attribute selection. When the program prompts the user to select an attribute, he responds by typing the attribute number. The program in turn replaces the selected attribute number by a dual asterisk. When all the selections have been made, the program requests a description of this new attribute set. In the example shown in B-5, Part B, the new name is "all waps." Table B-5, Part C, displays the last screen in this process.

After the completion of the first part of this option, the program then directs the user to the second part - the selection of the judges. The following information will be elicited from the user:

1. The context name
2. The judges name
3. The ID number of each judge  
(a 3 digit number the user assigns to a particular judge)
4. The name of the attribute set to be assigned to each judge

**Table B-5. Selection of Attribute Sets For Judging**

		Name	Mean	Std	Max	Min
A	01	SKT	66.00	4.50	95.00	0.00
	02	PFE	62.00	3.50	100.00	0.00
	03	TIS	36.00	3.20	66.00	24.00
	04	TIG	12.00	1.10	24.00	6.00
	05	AD	8.00	1.10	25.00	3.00
	06	IPR	48.00	0.02	50.00	1.00
	Enter number of attribute to put in attribute set ( $\emptyset$ to end) -->					
B	**	SKT	66.00	4.50	95.00	0.00
	**	PFE	62.00	3.50	100.00	0.00
	**	TIS	36.00	3.20	66.00	24.00
	**	TIG	12.00	1.10	24.00	6.00
	**	AD	8.00	1.10	25.00	3.00
	**	IPR	48.00	0.02	50.00	1.00
Enter number of attribute to put in attribute set ( $\emptyset$ to end) --> Enter description of this new attribute set -->						
C	**	PFE	62.00	3.50	100.00	0.00
	**	TIS	36.00	3.20	66.00	24.00
	**	TIG	12.00	1.10	24.00	6.00
	**	AD	8.00	1.10	25.00	3.00
	**	IPR	48.00	0.02	50.00	1.00
Enter number of attribute to put in attribute set ( $\emptyset$ to end) --> Enter $\emptyset$ to stop entering attribute sets else RETURN -->						

### Delete an Attribute Set

Any desired attribute set may be deleted under this option. The user will be required to input the specific context name and the name of the attribute set which is to be deleted. To prevent undesired deletions, the user is shown the names of both the context and the attribute set and must make a positive response for the identified attribute set to be deleted.

### Collect Judgments

This option requires the judge(s) or raters to rate each case that has been assigned to them. The program prompts for an identification of the desired context and a 3 digit identification number of the judge(s). The program then displays to the rater, one at a time, all the case numbers (profiles) with their computer generated statistics. The rater then assigns a value to that case and continues this process until all cases have been rated. An example of this procedure is shown in Table B-6.

Case Number 1	
SKT	64.81
PFE	63.53
TIS	42.54
TIG	11.08
AD	8.36
IPR	47.99
Enter your rating for this case -->	

### Compute Interrater Reliability

The program under this option computes, for a given context and attribute set, correlations for each rater. This enables the user to eliminate any aberrant rater before the regressions are computed in the next option. An example of the correlation computation may be seen in Table B-7.

<b>Table B-7. Interrater Reliability Analysis</b>	
<b>Interrater Reliability Analysis for:</b>	
<b>Context - 1. WAPS</b>	
<b>Attribute Set - 1. sam</b>	
<b>Judge ID</b>	<b>Correlation</b>
001	0.1441
002	0.4173
567	0.6278
565	0.6955
R11: 0.0488 Rkk: 0.1704	
Press <RETURN> to continue -->	

### Analyze Judgments

In order to analyze the rating policies of the individual judges and a joint policy which combines their individual policies, several descriptive statistics and equations are provided under this option. For example, the squared multiple correlation coefficient ( $R^2$ ) measures the percentage of the variance in the explained variable that is accounted for by the variance of all explanatory variables in the regression equation taken in combination. This statistic is displayed for each judge and for their combined policy. The mathematical weights derived from the program's analysis of the data determine the individual and group policies. These weights can then be used in subsequent applications of the policy capturing program to automate this "captured policy" to a new set of profiles. An example of this regression analysis may be seen in Table B-8.

**Table B-8. Regression Results**

**Regression Analysis for:**

**Context - 1. WAPS**

**Attribute Set - 1. sam**

Judge	001	002	567	565	ALL
RSQ	0.06	0.09	0.89	0.99	0.71
0	67.71	515.27	63.20	56.18	15.43
1	-0.15	-0.30	2.57	0.88	-0.24
2	0.11	-0.17	1.08	-1.09	0.04
3	0.05	-0.27	-1.68	-0.96	-0.12
4	3.83	0.60	18.21	6.93	4.41
5	-1.00	-1.17	20.09	-1.99	0.47
6	-1.04	-49.84	-12.40	0.04	0.24

**APPENDIX C**  
**SIMPLE MULTIATTRIBUTE RATING TECHNIQUE**  
**(SMART)**  
**USER'S MANUAL**  
**RELEASE 1.0**

## TABLE OF CONTENTS

	Page
I. BACKGROUND .....	56
II. ELICITATION TECHNIQUES .....	56
Model Form and Common Example .....	57
Elicitation of Single-Attribute Value Functions .....	57
Elicitation of Weights .....	59
Aggregation of Weights and Single-Attribute Values .....	62
III. GENERAL PRINCIPLES AND FEATURES .....	63
Using the Program .....	65
Construct Initial Problem/Solution .....	65
Decision Problem Options .....	65
Decision Tree Root Node .....	65
Decision Tree Sprouts .....	66
Decision Tree Weights .....	67
Twig/Option Range, Scores, and Location Measures .....	67
Report Options .....	68
Modification Options .....	71
Add/Delete Decision Tree Node(s) .....	72
Add/Delete Option(s) .....	73
Alter Importance Weights .....	73
Alter Location Measures .....	73
Sensitivity Analysis .....	74
Generate a Report .....	74

## LIST OF FIGURES

Figure		Page
C-1	Example of a Rating Scale for Assessing a Single-Attribute Value Function .....	58
C-2	Example of the Curve Drawing Technique to Assess a Single-Attribute Value Function .....	59
C-3	Illustration of a Hierarchical Tree Structure with Hierarchical Weights ...	61
C-4	Illustration of an Analytic Hierarchy .....	66

## LIST OF TABLES

Table		Page
C-1	Illustration of the Swing Weighting Technique .....	60
C-2	Illustration of Hierarchical Swing Weighting .....	62
C-3	Illustration of the Computation of Aggregate Value for a Promotion Candidate .....	63
C-4	Importance Weights for Sprouts .....	67
C-5a	Twig/Option Scores .....	68
C-5b	Twig/Option Scores .....	68
C-6	SMART/PC Report Options .....	69
C-7	Twig Importance Weights and Twig/Option Location Measures and Utilities .....	70
C-8	Aggregate Utilities .....	70
C-9	Problem Save File .....	71
C-10	Current Decision Tree Nodes .....	72
C-11	Add Decision Tree Node(s) .....	73
C-12	Sensitivity of Aggregate Utilities to Node Weights .....	74

## I. BACKGROUND

Edwards (1971) developed the Simple Multiattribute Rating Technique (SMART) as a direct response to Raiffa's (1969) article on multiattribute utility theory which Edwards found extremely stimulating but of limited practical usefulness because of the complexities in model forms and elicitation techniques. SMART was meant to capture the spirit of Raiffa's multiattribute utility procedures, while at the same time being simple enough to be useful for practical-minded decision makers. Through the years, Edward's procedure went through several metamorphoses, so that today SMART stands more for a collection of techniques rather than a single procedure (Von Winterfeldt & Edwards, 1986). The most recent versions of SMART are extremely close to the value measurement techniques but still retain much of the simplification spirit that motivated the early version.

## II. ELICITATION TECHNIQUES

In its simplest form, SMART uses direct rating and ratio weighting procedures for constructing utility functions (Gardiner & Edwards, 1975). First, scales are converted into value functions, either by rating the scale values (if scales are discrete) or by linear approximations (if scales are continuous). Next, attributes are rank ordered in the order of their importance. The lowest ranked attribute is given an importance weight of 10; the importance of the others is expressed in terms of multiples of 10. The resulting "raw" weights are normalized to add to 1. Because of the range insensitivity of importance weights (see Gabrielli & Von Winterfeldt, 1978; Keeney & Raiffa, 1976), recent SMART weighting methods have been changed to include "swing weighting," which is virtually identical to the weighting methods described in the value difference measurement models.

SMART applications have often used value trees, rather than building the multiattribute model simply on the level of the attributes. In tree applications of SMART, weights are elicited at all levels in the value tree and the final weights for attributes are calculated by "multiplying down the tree." This procedure has a number of advantages (see Stillwell, Von Winterfeldt, & John, 1987) as it facilitates the judgments and allows separation of weighting tasks in an organization between experts (lower level weights) and policy makers (higher level weights).

## Model Form and Common Example

The only model form that has been applied in the SMART context is the weighted additive model:

$$v(O_j) = \sum_{i=1}^n w_i v_i(X_{ij}).$$

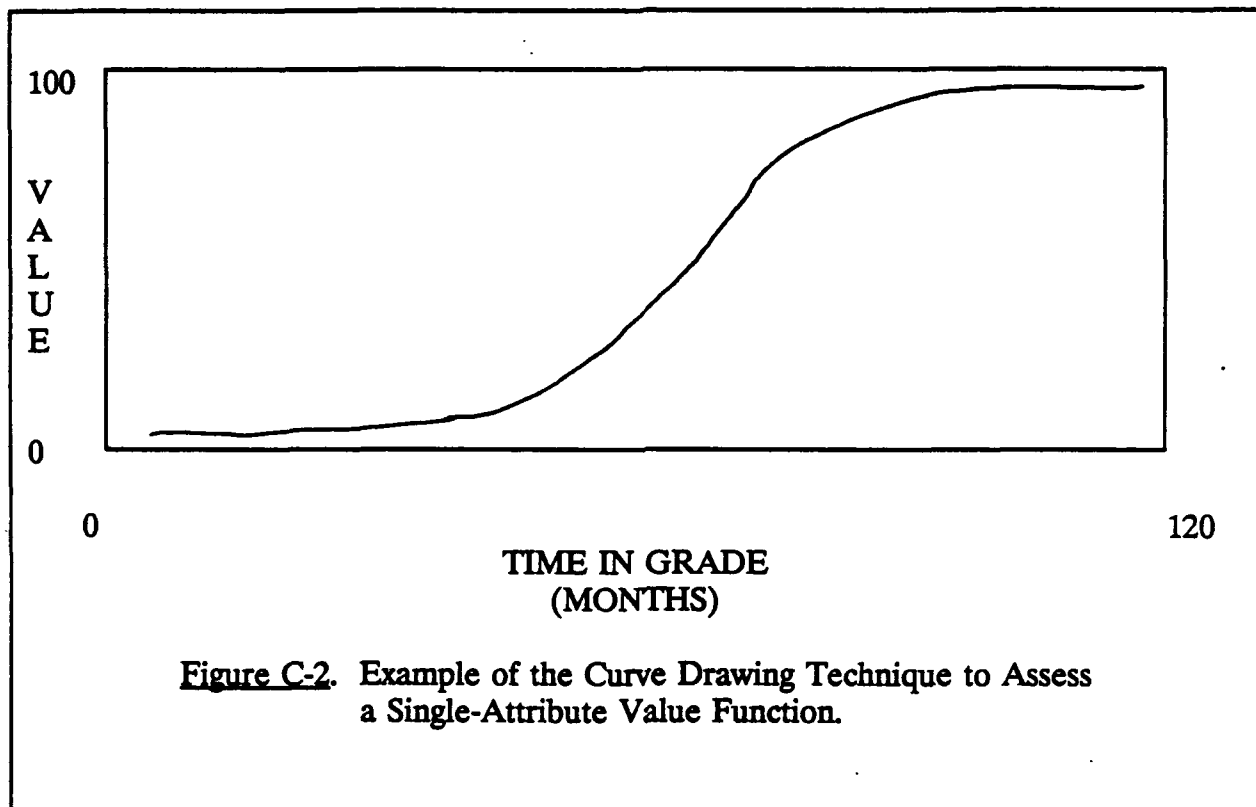
In order to clarify the exact form of the SMART technique that is described in this manual, the following sections provide a more detailed discussion of SMART, with an example of the application of that technique to a common personnel problem. The example presented is that of an Air Force enlisted promotion system discussed in Appendix B.

### Elicitation of Single-Attribute Value Functions

The first step in SMART involves development of single-attribute value functions which are constructed by arbitrarily assigning the "worst" level of a single-attribute scale a value of 0 and the "best" level a value of 100. Levels in between are rated on a continuous scale between 0 and 100, with instructions to consider carefully the difference in value between levels. If the underlying scale is numerical and continuous, curve drawing procedures are often substituted for this rating technique.

In the context of this promotion example, consider the attribute AD with several levels ranging from "no award or decoration" to the highest level consisting of numerous awards. As illustrated in Figure C-1, the level "no award or decoration" would receive a value of 0. The highest level, consisting of several examples of exemplary award combinations, would receive a value of 100. Next, the analyst would pick any of the intermediate levels on the scale, including individual awards and combinations of awards, and ask:





### Elicitation of Weights

Weights in SMART are assessed by the "swing weighting" method, in which the analyst presents the decision maker with a profile of a hypothetical alternative that has the worst level on each attribute and another hypothetical alternative that has the best level on each attribute. The decision maker is then asked to assume that he or she is "stuck" with the worst alternative, but has an opportunity to move one (and only one) attribute level from its worst to its best level. Which attribute would be most desirable to move? In other words, which change from worst to best level would add the most overall value in terms of determining the promotability of individuals? After identifying the attribute that provides this largest change or "swing," the decision maker identifies the attribute with the second largest change, the third largest, etc. This process provides a rank order of the weights in SMART.

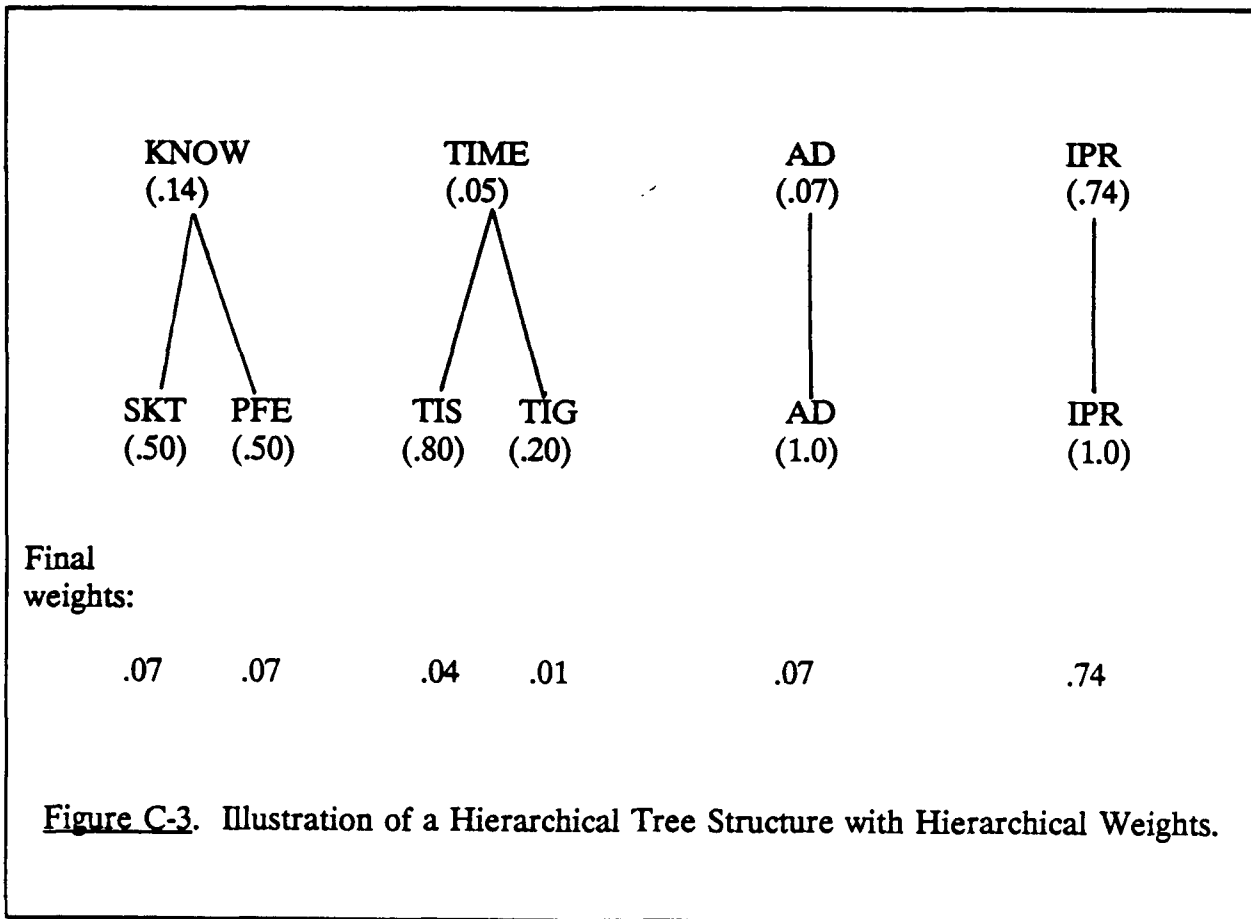
Next, the decision maker is asked to consider the value difference created by stepping from the worst to the best level in the most important attribute (i.e., the one that was chosen first), and to arbitrarily assign that value difference a score of 100. Similarly, an attribute for which the swing would make no difference in value at all is assigned a weight of 0. All other attribute swings are then given weights between 0 and 100. For example, an attribute that has the potential of adding half the overall value of the highest ranked attribute would receive a weight of 50. The resulting "raw" weights are summed up and each weight is divided by the total sum of the weights to create normalized weights that sum to one. When attributes are hierarchically structured, weights are assigned at each level of the hierarchy, and final attribute weights are obtained by multiplying the upper level weights by the lower level weights.

The swing weight method in the promotion example would be accomplished by asking the decision maker to rank order the desirability of moving an attribute from its worst to its best level. The decision maker might likely rank IPR score as the number 1 attribute, as a low IPR score would essentially make the candidate unpromotable. Following this change, the next most desired change may be in SKT, PFE, and AD, all of which may be considered to add approximately equal value to the promotion decision model. Next comes TIS, and TIG is last.

The swing in value in the IPR attribute would then be given a weight of 100 points. All other weights are expressed in values between 0 and 100. Hypothetical results are shown in column 3: These raw weights are highly skewed, because the IPR attribute produces an extreme swing in value (in practice one might worry about the definition of the endpoints of that scale, or refine this attribute by breaking it down into subattributes). Normalization of these weights is done mechanically. At the bottom of column 3 of Table C-1 is the sum of the raw weights and in column 4 are the normalized weights, which, of course, total 1.00.

Table C-1. Illustration of the Swing Weighting Technique			
Attribute	Rank of Swing	Raw Weight of Swing	Normalized Weight
SKT	2	10	.07
PFE	2	10	.07
TIS	3	5	.04
TIG	4	1	.01
AD	2	10	.07
IPR	1	100	.74
		sum: 136	sum: 1.00

To illustrate hierarchical weighting, consider the tree structure in Figure C-3. In this case it might be logical to first weight SKT versus PFE with respect to the knowledge (KNOW) objective only, then to weight TIS versus TIG with respect to the time (TIME) objective only. This can be done with the swing weighting procedure exactly as described above, and it would produce the results indicated in Table C-2, section 2a. Next, weighting of the relative swings of the four higher level objectives KNOW, TIME, AD, and IPR is done by asking the decision maker to simultaneously consider swings of attributes under the objectives that are to be weighted. A specific question might be: "Would you rather change both SKT and PFE from their worst levels to their best levels or change both TIG and TIS from their worst to their best levels?" The answer to this question would provide a rank order of the weights for KNOW and TIME. The questions regarding the other two attributes (AD and IPR) would be identical to those illustrated in the non-hierarchical case. Together they might provide a rank order as shown in Table C-2, section 2b. Raw and normalized weights are also shown in that table. The final weights for the lower level attributes SKT, PFE, TIS, and TIG are obtained by multiplying the upper normalized weight with the respective lower level normalized weight (see Figure C-3).



**Table C-2. Illustration of Hierarchical Swing Weighting**

	Attribute	Rank of Swing	Raw Weight	Normalized Weight
<b>2a (Lower level)</b>				
KNOW				
	SKT	1	100	.50
	PFE	1	100	.50
TIME				
	TIS	1	100	.80
	TIG	2	20	.20
<b>2b (Upper level)</b>				
KNOW				
	SKT	2	20	.14
	PFE			
TIME				
	TIS	4	6	.05
	TIG			
	AD	3	10	.07
	IPR	1	100	.74

**Aggregation of Weights and Single-Attribute Values**

The aggregation of weights and single-attribute values is accomplished as follows. For each alternative  $O_j$ , a profile of attribute levels  $X_{ij}$  is generated which indicates the degree to which that alternative scores on the attributes. The  $X_{ij}$ 's are converted into single-attribute values  $v_i(X_{ij})$  which are simply read off the value curves and graphs as shown in Figure C-1 and C-2. The overall value of the alternative is then calculated by the formula:

$$v(O_j) = \sum_{i=1}^n w_i v_i(X_{ij}).$$

In the promotion example, a promotable candidate may have the profile described in column 2 of Table C-3. The associated single-attribute values and weights might be as shown in columns 3 and 4. Multiplying weights and single-attribute values generates column 5, and adding these cross-products produces the overall value of 71.95 for this candidate.

Table C-3. Illustration of the Computation of Aggregate Value for a Promotion Candidate				
Attribute	$X_{ij}$ Candidate 0's Scoring Profile	$V_i X_{ij}$ Relative Single-Attr. Values	$W_i$ Weights of Attr.	$W_i V_i X_{ij}$ Cross-Products
SKT	50 points	50	.07	3.50
PFE	75 points	75	.07	5.25
TIS	60 months	50	.04	2.00
TIG	12 months	25	.01	.25
AD	AF Commend	25	.07	1.75
IPR	100 points	80	.74	59.20
			Total value:	71.95

### III. GENERAL PRINCIPLES AND FEATURES

SMART/PC is a tool for conducting a systematic evaluation of two or more multi-attributed options or actions. SMART/PC facilitates the following tasks:

- o Define the options/actions for evaluation.
- o Define the evaluative criteria for the options by building a decision tree consisting of higher level general objectives down to the specific attributes or "twigs" which achieve or represent these objectives.
- o Weight all nodes in the tree for importance.

- o Locate each option on the bottom level twigs.
- o Derive single-attribute utilities and an overall "aggregated" utility for each option.
- o Perform sensitivity analyses by varying importance weights and location measures.

The following "rules of thumb" apply to the operation of the program unless otherwise stated.

- o All keyboard entries must be in UPPER CASE.
- o Only one keyboard entry may be made at a time.
- o All program entries except function key entries are input by typing in the appropriate characters followed by the RETURN key.
- o Default values can often be entered by typing only the RETURN key.
- o Program prompt - User reply transactions are mediated through the use of special prompt screens.
- o All of the prompt screens allow for a "help" request for an explanation of what is being prompted. The "F1" function key conveys this request.
- o Some of the prompt screens allow for other special requests which are listed below:

"End"	--> end this prompt session - keep the changes
"RETURN"	--> keep the current value or use the default
"Esc"	--> end this prompt session - cancel this entry

## Using the Program

The following options will be covered in this manual:

- o Construct Initial Problem/Solution
- o Revise Existing Problem

### Construct Initial Problem/Solution

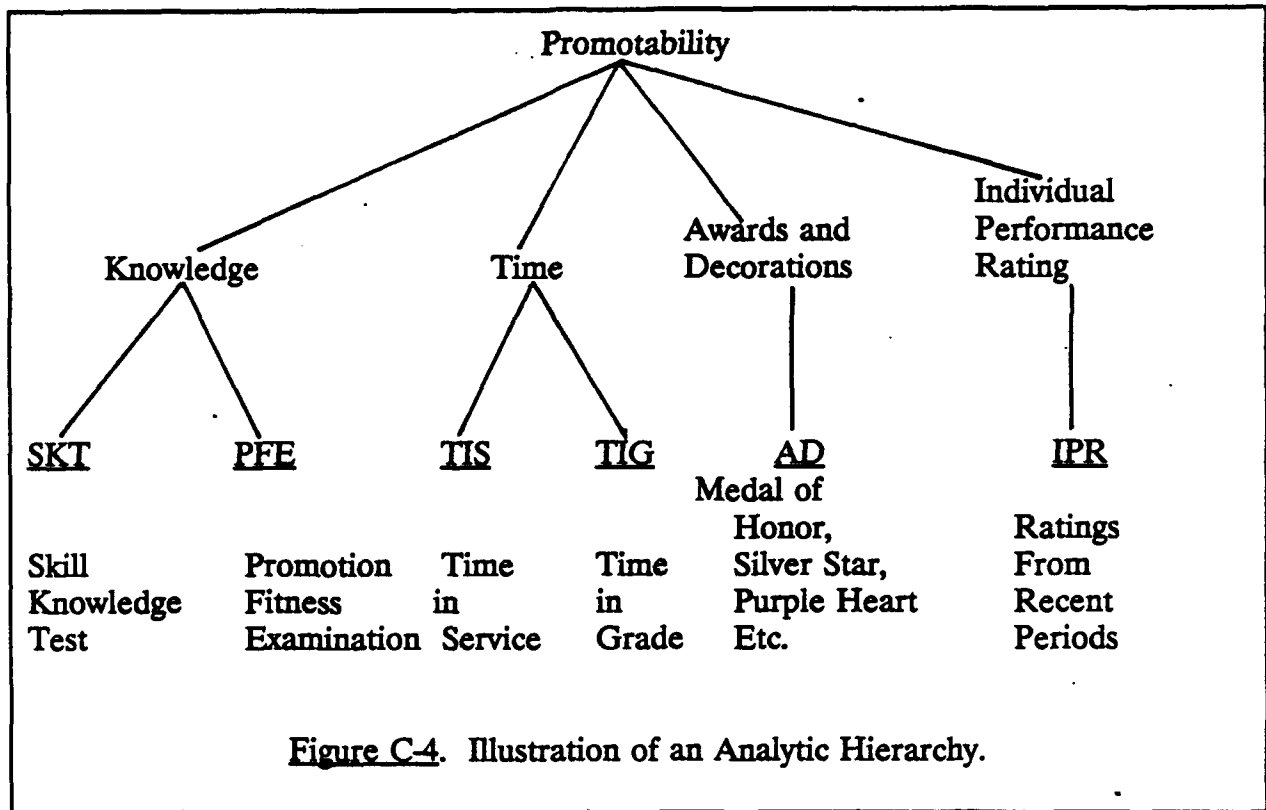
#### Decision Problem Options

After the user indicates a desire to construct an initial problem/solution, SMART will request the user to enter an option for evaluation. Options are the set of alternatives which the user will be evaluating for overall desirability or utility through the use of the SMART procedure. These options may represent judgments about objects, actions, people, programs, etc. After each option has been entered the "RETURN" key should be pressed and the next option entered. After all the options have been entered, the "END" key should be pressed signifying that all options have been entered.

The program then requests a Y or N response to the following question, "Is a Problem Structuring Module (PSM) Decision Tree File to be used for this session? The PSM program is a component of the Integrated Decision Modeling System and enables a user to build a hierarchy for analyzing a decision problem. Assume, however, that the user did not want to use the PSM Module and he responded with an N.

#### Decision Tree Root Node

The user is prompted for a description of the overall decision problem. This description will be the "Root Node" for the decision tree. From this starting point the user can structure the problem at evaluating various decision options by defining a hierarchy of nodes which flow from the "Root Node." The hierarchy proceeds from more general properties or objectives of any given option, down to the specific measurable attributes which characterize a set of options. An example of this analytic hierarchy is displayed in Figure C-4.



### Decision Tree Sprouts

The user must now enter a description of a node which is a "Sprout" to the "Root Node" previously entered. In Figure C-4, Promotability is a parent node to Knowledge, Time, Awards and Decorations, and Individual Performance Rating. Also, those four are each parent nodes to the twigs below them. For example, Knowledge is a parent node to both SKT and PFE. For this implementation of the SMART procedure any sprout can have only one parent, but a parent may have more than one sprout.

In entering sprouts like those displayed in Figure C-4, the process would be as follows: SMART would prompt the user to enter a description of a node which is a "Sprout" to the parent node "Promotability." The user would then enter, one at a time, the sprouts Knowledge, Time, Awards and Decorations, and Individual Performance Rating; after the last of these had been entered, the user would press the "RETURN" key to signify that all the sprouts on the first level had been entered. SMART would again prompt the user to enter a description of a node which is a sprout to the parent node Knowledge. Following the procedure just outlined, the user would enter SKT and then PFE. After PFE, the user would press the "RETURN" key and the program would then ask for sprouts to the parent node Time. This process would be continued through the parent node Individual Performance Rating. When the last twig, IPR, has been entered, the user then presses the "END" key to move to the next step in the program.

## Decision Tree Weights

The decision tree nodes are weighted on the basis of importance. All nodes at one level of the decision tree are weighted with respect to each other. This may be accomplished in one of several ways. For example, the least important node at a given level of the tree might receive a weight of "10." The other nodes at that level receive weights which reflect how much more important they are than the reference node. Another way to apply the weights is to assign a weight of "100" to the most important node at each level of the decision tree and to weight the other nodes accordingly.

The decision tree displayed in Figure C-4 will be used to illustrate the process of entering the decision tree weights. First, the program will prompt the reader by displaying a screen as shown in Table C-4.

<u>Table C-4. Importance Weights for Sprouts</u>	
Enter importance weights for the sprouts of the parent shown below (most important = 100).	
PARENT: Promotability	
.....	Knowledge
.....	Time
.....	Awards and Decorations
.....	Individual Performance Rating
F1 (Help)	

After the user has entered the requested weights, the program screen would now show Knowledge as the parent node and would request weights for sprouts SKT and PFE. This process would be continued until all the parent nodes on this level would have weights assigned to their sprouts.

## Twig/Option Range, Scores, and Location Measures

During this prompt session the user will be asked to provide the maximum and minimum values or scores that a given option can have for the twig shown on the screen. In addition, the user must respond to the question "Is More Better" with respect to the Twig. After providing a scoring range for the twig and the direction of the utility function, the user will then provide a score for each option with respect to the displayed twig. These scores will be converted into location measures which range from "0" (lowest utility) to "100" (highest utility), by way of a linear fitting equation.

The twigs shown on the lowest level of Figure C-4 will be used to demonstrate this procedure. Initially the user would be shown a screen like that shown in Table C-5a. The user would then enter the minimum and maximum values for that twig and indicate if more was better. After entering the information and pressing the "RETURN" key, the user would then be shown a screen like Table C-5b. Here the user would respond with a score for each option being considered. If the options were the set of airmen being considered for promotion, then each option might be referred to as A1, A2, ... An. The user would score the first option, press the "RETURN" key, score the second option, and continue on until all options have been given a score for that twig. The procedure would continue in the manner just described until each twig had been given minimum and maximum values and the direction of the utility function indicated; also, a score would be entered for each option with respect to every twig.

Table C-5a. Twig/Option Scores
Twig: SKT
Minimum Value _____
Maximum Value _____
Is More Better? (Y/N): _____

Table C-5b. Twig/Option Scores
Twig: SKT
Option: A1
Score _____

Report Options

The program is now ready to provide the user with the various reports as shown in Table C-6.

**Table C-6. SMART/PC Report Options**

Indicate the report options of your choice by entering a "Y" in the space next to the option.

Decision Tree Nodes	.....
List of Options	.....
Importance Weights	.....
Location Measures	.....
Twig Utilities	.....
Aggregate Utilities	.....

F1 (Help)

From the table it can be seen that SMART can output a list of the nodes of the decision tree. The root, primary, secondary, and twig nodes will all be listed and nodes of the same type will be grouped together in the same indentation block. Decision options being evaluated are listed and shown in tabular form. Twig importance weights, twig/option location measures, and twig/option utilities are computed by programs within SMART and displayed in various tables. A combination of these tables is shown in Table C-7.

**Table C-7. Twig Importance Weights and Twig/Option Location Measures and Utilities**

TWIG IMPORTANCE WEIGHTS	
KT	.070
FE	.070
IS	.040
IG	.010
DA	.070
RPI	.740

---

TWIG/OPTION LOCATION MEASURES	
KT	
A2	
	55.000

---

TWIG/OPTION UTILITIES	
FE	
A1	
	5.250

Table C-8 displays a utility for each option. An illustration of the computation of an aggregate value for each utility may be reviewed by viewing Table C-3.

Table C-8. Aggregate Utilities	
A1	61.750
A2	66.320
A3	79.250
A4	79.200

It is this aggregate value that is used in comparing the overall merits of the options.

After the aggregate utilities report has been completed, the program then gives the user an opportunity to create a "Problem Save File." Information comprising the file is displayed in Table C-9. The user is also asked to provide a name for this file for future reference.

**Table C-9. Problem Save File**

The problem parameter save file contains all of the values which describe a decision problem and a solution to it. The save file contains the following parameters.

- No. of options/No. of nodes
- Decision Options
- Decision Tree Nodes
- Node Status Indicators
- Node Parent Pointers
- Node Sprout Pointers
- Raw node wts., normalized wts., twig wts.
- Scoring range & more/less indicator
- Location Measures
- Twig Utilities
- Aggregate Utilities

**Modification Options**

After entering the name of the problem parameter save file, the user can modify the initial problem by adding or deleting decision tree nodes and decision options, and by altering importance weights and location measures. A sensitivity analysis may also be accomplished. The keys and their associated option are as follows:

- |    |                                  |
|----|----------------------------------|
| F3 | Add/Delete Decision Tree Node(s) |
| F4 | Add/Delete Option(s)             |
| F5 | Alter Importance Weights         |
| F6 | Alter Location Measures          |
| F7 | Sensitivity Analysis             |
| F8 | Generate a Report                |

### Add/Delete Decision Tree Node(s)

The user may modify the current representation of the decision problem by adding new nodes to the existing decision tree. Each new node must be specified with respect to a currently existing parent node. The user can obtain the current list of nodes, their descriptions, types, and identifying numbers, by typing "F10" instead of a parent node number. When the user is finished viewing the list of current nodes, the user will be able to resume the add-node session where the user left off. Table C-10 reflects the information displayed on the current decision tree nodes and Table C-11 shows the prompt the program gives the reader when adding a new node.

<b>Table C-10. Current Decision Tree Nodes</b>			
The current decision tree nodes are shown below along with identifying number, type indicator, and parent id number.			
<b>ID Number</b>	<b>Node Description</b>	<b>Node Type*</b>	<b>Parent ID No.</b>
1	PROMOTE	R	
2	KNOW	S	1
3	TIME	S	1
4	AD	S	1
5	IPR	S	1
6	SKT	T	2
7	PFE	T	2
8	TIS	T	3
* R = Root Node S = Sprout Node T = Twig Node			

**Table C-11. Add Decision Tree Node(s)**

You may add a node to your current decision tree by entering the number of the node which will be the parent node for your new node. In addition, type in the name of your new node.

Parent Node      (RETURN key --> List Nodes)  
Number

New Node \_\_\_\_\_

F1 <Help>

End <Done>

The process to delete one or more nodes from a current decision tree is accomplished in a similar manner except the user deletes rather than adds. Also, the addition or deletion of decision tree nodes makes it necessary to input some new importance weights. The program will prompt the user for these values.

#### Add/Delete Option(s)

The user may now indicate if options are to be added or deleted by indicating either A or D and entering the name(s) of the options. In the case where new options have been added, the user will be prompted to provide a score for each option with respect to every twig.

#### Alter Importance Weights

This option allows the user to review/modify importance weights assigned to the various attributes. All sprouts and twigs are displayed and their importance weights may either be changed or their current value retained.

#### Alter Location Measures

The user may alter the location measures by changing the minimum/maximum value for each twig and also the direction of its utility function. He is also shown the score for each option with respect to every twig.

### Sensitivity Analysis

Under this feature, the user may examine how changes in the weight for any given node, affect the aggregate utilities of the current set of options. An example of this is shown in Table C-12.

Table C-12. Sensitivity of Aggregate Utilities to Node Weights									
NODE: IPR (.79)									
Options	.1	.2	.3	.4	.5	.6	.7	.8	.9
A1	71.7	73.5	75.3	77.1	78.9	80.7	82.5	84.3	86.1
A2	88.5	88.2	88.0	87.7	87.4	87.1	86.8	86.5	86.2
A3	62.4	64.2	65.9	67.6	69.3	71.1	72.8	74.5	76.2

### Generate a Report

The last feature under the "Revise Existing Problem" section of SMART allows the user to generate various reports relative to the modified program. This procedure has been covered in the report options section of this manual.

**APPENDIX D**

**HIERARCHICAL ADDITIVE WEIGHTING METHOD  
(HAWM)**

**USER'S MANUAL**

**RELEASE 1.0**

## TABLE OF CONTENTS

	Page
I. INTRODUCTION .....	77
Features .....	77
Example .....	78
II. ELICITATION OF WEIGHTS .....	79
III. PREFERENCE SCORES .....	80
IV. AGGREGATION RULE .....	82

## LIST OF FIGURES

Figure	Page
D-1 Illustration of an Analytic Hierarchy .....	78
D-2 Illustration of the HAWM Aggregation Process .....	82

## LIST OF TABLES

Table	Page
D-1 Illustrative Weight Ratio Assessment .....	80
D-2 Illustration of Relative Preference Assessments for Five Promotion Candidates on the "Decoration" Attribute .....	81

## I. INTRODUCTION

The HAWM is a multiple attribute level decision making (MADM) method. The first level must have a single objective. (a plane, a job, overall wellbeing, etc.). The bottom level is always the alternatives considered. (plane A, B, C. . etc). Middle levels are attributes about the bottom level of alternatives.

### Features

Max Levels	=	5
Max Items per Level	=	7
Item Name Length	=	10

The names are all entered consecutively. When finished entering names for the current level, a single carriage return begins prompting for the next level of names. When all the levels have been entered, return terminates the name entry session. After the names have been entered, the pairwise comparison matrices are entered.

HAWM is menu driven and consists of the following options:

1. Load new dataset
2. Run Hierarchial Additive Weighting Method
3. Display Results
4. Edit input data
5. Save current dataset
6. Configure system
7. Quit program

1. Load new dataset: The data that HAWM uses consists of alternative or attribute names for each level of the hierarchy and a set of pairwise comparisons for each adjacent level. Data can be entered from the keyboard or loaded from a previously made data file. If you request file input, all the files with the extension of .EIG will be displayed on the screen.

2. Run HAWM: Executes the HAWM algorithm determining the weighting of each alternative and attribute.

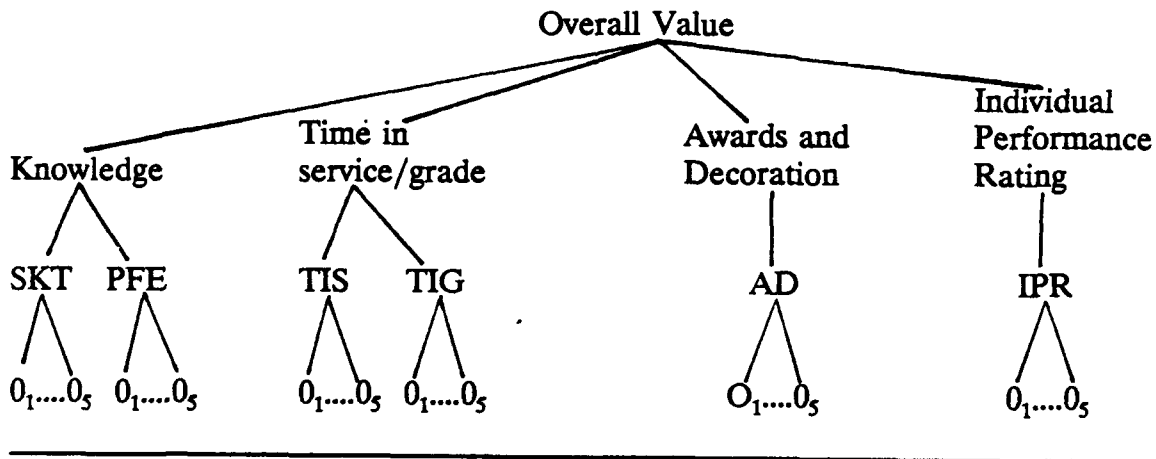
3. Display Results: Output the results of the HAWM algorithm to printer, screen, or disk.

4. Edit input data: Allows the data entry operator to edit the data and verify it for correctness.
5. Save current dataset: Saves the data that has been entered for later use.
6. Configure system: Allows the operator to select the color of the screen display, the width of the printer, the number of decimals printed, etc. The changes made in this menu will be saved in a configure file and used the next time HAWM is run. This allows the operator to select colors and formats that are the best for him and use them later as default values.
7. Quit: Exits from HAWM and returns operator to DOS.

### Example

In order to clarify the exact form of the HAWM technique that is described in this manual, this section provides a more detailed discussion of HAWM, with an example of the application of a common personnel problem. The example presented is that of the Air Force enlisted promotion system, discussed in Apendices B and C.

HAWM begins with an hierarchical structure of the evaluation problem, with top values that are very much like a SMART structure. However, at the bottom, the alternatives fan out under each attribute as yet another level of evaluation in the tree. Figure D-1 presents the HAWM analogue for the promotion example. Here promotable airmen are the alternatives ( $O_j$ ) and are repeated at the bottom of the tree.



**Figure D-1.** Illustration of an Analytic Hierarchy.

## II. ELICITATION OF WEIGHTS

The HAWM process begins by eliciting weights in the upper part of the tree. Weights are also elicited for the bottom level (the alternatives) to indicate their relative desirability in achieving bottom level objectives or attributes. Since that step is somewhat similar to the value function assessment in other procedures, it will be discussed in a separate section.

In the upper part of the tree, weights are interpreted to reflect the "relative importance" of the attributes or objectives. Weights are assessed under each node, comparing possible attributes with pairwise weight judgments. The decision maker is presented with one pair and asked:

- 1) Which attribute do you think is more important?
- 2) On a scale from 1 to 9, how much more important is that attribute (1 meaning equally important, 9 meaning much more important)?

The numbers obtained from these weighting judgments are considered weight ratios and entered into an  $n \times n$  (attributes by attributes) matrix of weight ratios in which the diagonals are set to 1. In the HAWM, the additional assumption is made that the weight ratios must be reciprocal. Thus, a set of  $n(n-1)/2$  weight ratios fills the complete  $n \times n$  matrix that defines weight ratios at each node.

Having obtained  $n(n-1)/2$  weight ratios, the HAWM solves for the "best fitting" set of normalized weights, i.e., those weights that can best reproduce the (possibly inconsistent) assessed weight ratios. The HAWM solves these weights as the eigenvector of the weight ratio matrix. In addition to providing the best fitting weight solution, the HAWM also provides an index of (in) - consistency which ranges from 0 (perfect consistency) to 1 (highly inconsistent weight ratio assessments).

In the promotion example, the upper level weight ratio assessment might produce a weight ratio matrix as the one shown in table D-1. The circled numbers are the assessed ones. The others are inferred from the reciprocity assumption. The diagonals are simply assumed. The last column of table D-1 shows the weights derived from the HAWM program (as run in the HAWM software) and indicates that there is moderate consistency in the weight ratio assessments. After such an initial assessment, the decision maker is asked if the ratios should be revised or kept unchanged.

**Table D-1. Illustrative Weight Ratio Assessment**

	Know	Time	AD	IPR	Normalized weights
Know	1	3	2	1/9	.13
Time	1/3	1	1/2	1/9	.06
AD	1/2	2	1	1/9	.09
IPR	9	9	9	1	.72

Inconsistency score: .054

---

If satisfied with the current assessment, the decision maker goes on to lower level nodes of the value tree repeating the process described above. In the example, there are only two lower level nodes: SKT vs. PFE and TIS vs. TIG. The decision maker is asked to provide relative weight ratios for each of these pairs considering the contribution to achieving the next higher objective (KNOW or TIME). Both weight assessments would generate 2x2 matrices with no possibility for inconsistencies. For example, the assessed weight ratio of SKT vs. PFE might be 2 resulting in relative weights of .67 for SKT and .33 for PFE. Similarly, the assessed weight ratio for TIS vs. TIG might be 3 resulting in relative weights of .75 for TIS and .25 for TIG. Since there exists no possibility for inconsistency, the results are identical to those obtained by simply normalizing the raw weight ratios.

### III. PREFERENCE SCORES

Once the bottom level of alternatives is reached, the decision maker has two choices in HAWM. Either continue the judgments of relative importance or produce judgments of the relative preference of the alternatives with respect to achieving the lowest level attribute. Since in the context discussed here, the latter interpretation is more intuitive, only this variant of the HAWM will be discussed.

Under each lowest level node, and for each pair of alternatives, the decision maker is asked:

- 1) Which of the two alternatives do you prefer with respect to the attribute under consideration;

- 2) On a scale from 1 to 9 (1 meaning indifference, 9 meaning extreme preference), how much do you prefer this alternative on the attribute under consideration?

As in the importance weight assessment, the relative preference assessments are assumed to be reciprocal, so that  $n(n-1)/2$  assessments are sufficient to fill out the complete  $n \times n$  matrix. The final scores for each alternative are again the eigenvector of that matrix that best matches the relative preference ratios.

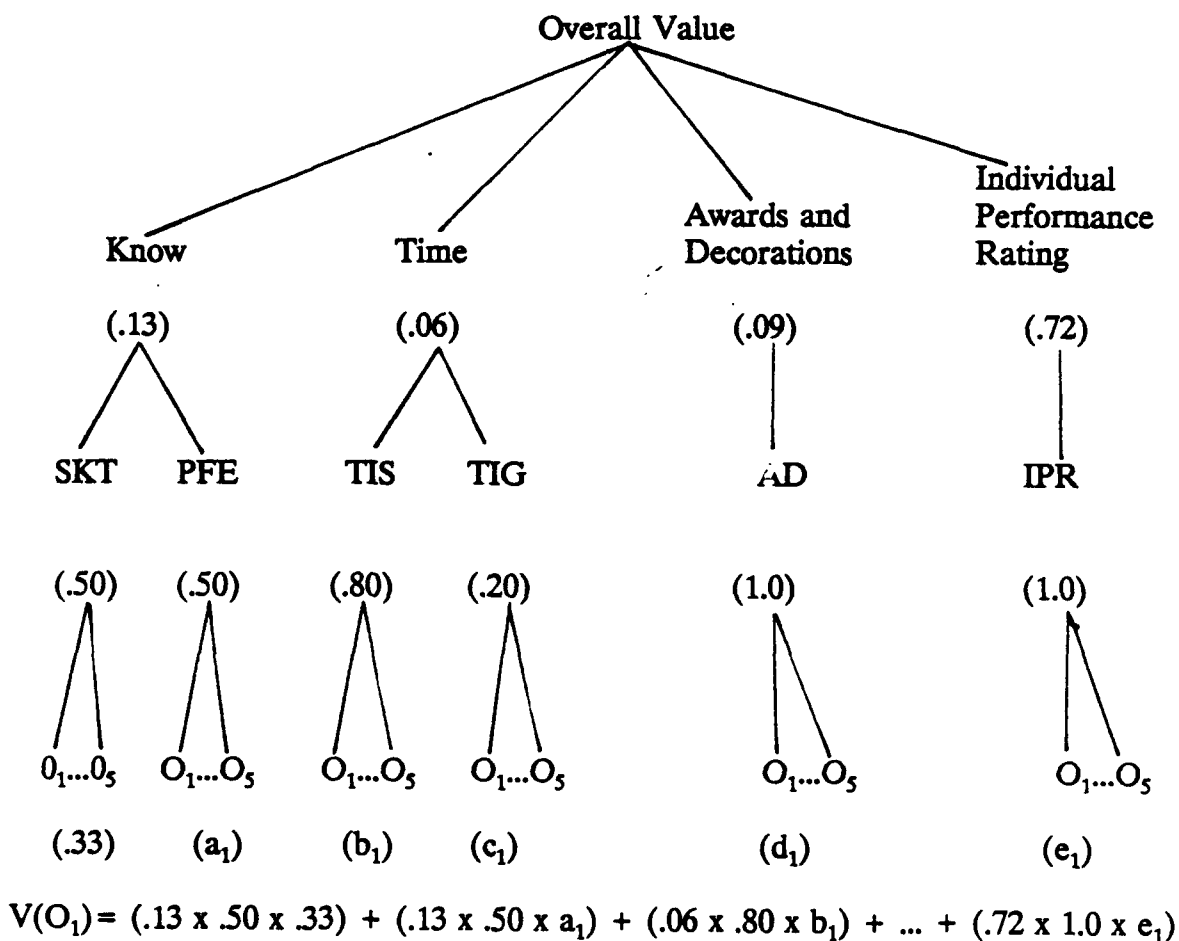
To illustrate this process in the promotion context, consider the attribute AD and assume that 5 promotable candidates have different levels of that attribute. A preference comparison of these five airmen might look like the one in Table D-2. The last column in that table indicates the renormalized scores that each of the candidates receives as a result of the relative preference judgments. The index at the bottom of the table shows the assessments were somewhat inconsistent based on the scale of 0 (perfect consistency) to 1 (highly inconsistent).

**Table D-2. Illustration of Relative Preference Assessments for Five Promotion Candidates on the "Decoration" Attribute**

	Candidate					Relative Score
	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	A <sub>4</sub>	A <sub>5</sub>	
A <sub>1</sub>	1	6	3	2	1	.33
A <sub>2</sub>	1/6	1	1/2	1/3	1/6	.05
A <sub>3</sub>	1/3	2	1	2	1/3	.14
A <sub>4</sub>	1/2	3	1/2	1	1/2	.14
A <sub>5</sub>	1	6	3	2	1	.33
Inconsistency score: .034						

#### IV. AGGREGATION RULE

The results of weighting and preference assessments are aggregated in a form that is very similar to the SMART rule by "multiplying down" the tree and adding the multiplicative elements for each alternative. Consider, for example, the assignments of relative weights and preference judgments in Figure D-2. The overall value of alternative  $O_1$  would be calculated by multiplying down all the normalized scale values for that alternative. Thus, for example, the weight on KNOW (.13) would be multiplied by the weight on SKT (.5) which in turn would be multiplied by the preference score of candidate  $O_1$  on attribute SKT (.33). Having done similar calculations for each of the paths connecting the top of the tree with alternative  $O_j$  at the bottom, the analysis then simply adds these crossproducts to generate the overall evaluation of alternative  $O_j$ . The overall values of the other alternatives are calculated in a similar way.



**Figure D-2.** Illustration of the HAWM Aggregation Process.

**APPENDIX E**  
**SENSITIVITY ANALYSIS MODULE**  
**USER'S MANUAL**  
**RELEASE 1.0**

## TABLE OF CONTENTS

	Page
I. INTRODUCTION .....	85
II. PAYGEN STARTUP .....	85
PAYGEN Parameters .....	85
Variable Selection .....	87
Function Selection .....	89
Main Menu .....	90
III. ERROR HANDLING .....	91
APPENDIX E-1: SAMPLE INPUT DATA FILE .....	92
APPENDIX E-2: SAMPLE OUTPUT DATA FILE .....	93
APPENDIX E-3: SAMPLE REPORT .....	94
APPENDIX E-4: PAYGEN ERROR MESSAGES .....	97

## LIST OF FIGURES

Figure	Page
E-1 Screen 1 - PAYGEN Parameters Parameters .....	86
E-2 Screen 2 - Variable Selection .....	88
E-3 Screen 3 - Function Selection .....	89
E-4 Screen 4 - Main Menu .....	90

## I. INTRODUCTION

To perform sensitivity analysis a program called payoff generator (PAYGEN) developed at the Air Force Human Resources Laboratory, Brooks Air Force Base, Texas is used. PAYGEN is used in conjunction with the policy specifying program (POLSPEC) and applies a policy to data and generates the data's associated payoffs.

PAYGEN is user friendly requiring the user to select the payoffs needed to generate. PAYGEN will then generate a report containing payoffs and statistics on those payoffs. The statistics consist of the following: lowest and highest value, mean, standard deviation, popular standard deviation, and correlations between payoffs. PAYGEN will also generate an output payoff file consisting only of selected payoffs.

PAYGEN has a view module that will allow you to look at any standard ASCII text file. A print module is also provided to obtain a printout of a generated report, output payoff file, or any other ASCII text file. An editor is available to change the values contained in the input data file.

PAYGEN will also keep track of all the file names used by creating a parameter file. All the user needs to enter is the general name of the policy and PAYGEN will do the rest. It will also save the output headings used to describe your policy.

## II. PAYGEN STARTUP

### PAYGEN Parameters

Parameter Screen 1 will be the first screen displayed. An example of this screen is found in Figure E-1. It consists of all the file names needed to run PAYGEN. At startup the cursor will be by the parameter file. Enter a general name with a drive specifier that describes the policy. If the POLSPEC policy is called SAMPLE and the files are on Drive B:, enter "B:SAMPLE". PAYGEN will fill in the remaining file names with the default extensions. These are by no means set. You may move the cursor to any one of these lines and change the associated name. At the end of the session the user is able to write the information on this screen to the parameter file.

Each of the files listed in screen 1 either contain necessary data needed to run PAYGEN, or will be created as a product of PAYGEN. The file descriptions are as follows:

**Parameter File:** The Parameter file will be created by PAYGEN, and will contain all the information contained in screen 1. This includes



- Input Record Limit:** The Input Record Limit is used for testing prior to making the final PAYGEN run. This sets a limit on the number of records read from the Input Data File. The initial setting is "ALL", meaning that all the records from the Input Data File will be read. On large files where many payoffs need to be calculated, set this to a small value, until the data is formatted correctly.
- Variable Name File:** The Variable Name File is a file generated by POLSPEC containing the variable identifications and variable descriptions used in the policy function file. The default extension used on this file is .NAM.
- Policy Function File:** The Policy Function File is the file generated by POLSPEC containing the parameters needed to generate a function equation. PAYGEN will read this file and apply the functions to the values in the Input Data File. This file should not be altered by the user as it could possibly change the equations that will be used in generating the payoff value. The default extension for this file is .POL.
- Output Data File:** The Output Data File is created by PAYGEN, and contains the generated payoff values. See Appendix E-2 for a sample file. The file contains 1 record for each record read from the Input Data File. The format of this file is determined by the user when the payoff selections are made. The file has a maximum record length of 255 characters. This should be taken into account when selections are being made. The default extension for this file is .DAT.
- Output Report File:** The Report file will be created by PAYGEN, and will contain the PAYGEN report. See Appendix E-3 for an example of this report. The Output Headings are entered by the user. They are used on the PAYGEN report, and on any printout made by PAYGEN.

### Variable Selection

After all the data for screen 1 is filled in the user proceeds to screen 2. To move to screen 2 press the PgDn key once. Screen 2 will take a second or two while it reads the variable identifications from the variable names file. When it does display the information, it will display the variable name, print flag, output format and description. See Figure E-2 for an example of screen 2. The user starts with the format for the first variable will be highlighted. This signifies that the user may change the output format of that variable.

The variables as well as the function payoffs may be output to the Output Data File. If the variable is should not be output to the data file, change the format to F0.0. This can be done by simply pressing the F2 key. The F2 key acts as a toggle between F0.0 and F5.0. To choose a format other than F5.0, simply press the "F" key. A cursor will appear in the highlighted output format, and the user may change it to some other value. To change the output format to F8.3 simply type "F8.3". The up or down arrows will move through the different output formats.

PAYGEN v860903	VARIABLES	Parameter Screen 2
<u>NAME</u>	<u>PRN - FMT</u>	<u>DESCRIPTION</u>
X01	* F5.0	VARIABLE 1
X02	* F5.0	VARIABLE 2
X03	* F5.0	VARIABLE 3
X04	* F5.0	VARIABLE 4
X05	* F5.0	VARIABLE 5
X06	* F5.0	VARIABLE 6
X07	* F5.0	VARIABLE 7
X08	* F5.0	VARIABLE 8
X09	* F5.0	VARIABLE 9
X10	* F5.0	VARIABLE 10
X11	* F5.0	VARIABLE 11

Avail Keys: Esc, PgUp, PgDn, F1 = Help, F2 = Toggle, F = Update

Figure E-2. Screen 2 - Variable Selection.

The PRN column on screen 2 is simply a flag column that shows which variables will be output to the data file. The flag is set automatically whenever the output format is other than F0.0.

### Function Selection

After variables for screen 2 have been read in, and the output formats are selected, the user can move on to screen 3 by pressing the PgDn key. Before displaying screen 3, the program will read in the functions from the policy function file. PAYGEN will display on the screen which function it is reading.

PAYGEN v860903			FUNCTIONS		Parameter Screen 3
<u>NAME</u>	<u>PRN - FMT</u>	<u>PARAMETERS</u>	<u>MODEL</u>	<u>DESCRIPTION</u>	
F01	* F6.1	X01 , X02	1	FUNCTION 1	
F02	* F6.1	X01 , X02	1	FUNCTION 2	
F03	* F6.1	F01 , F02	1	FUNCTION 3	
F04	* F6.1	X03 , X04	1	FUNCTION 4	
F05	* F6.1	F03 , F04	1	FUNCTION 5	
F06	* F6.1	X05 , X06	1	FUNCTION 6	
F07	* F6.1	F05 , F06	1	FUNCTION 7	
F08	* F6.1	X08 , X07	1	FUNCTION 8	
F09	* F6.1	F08 , F09	1	FUNCTION 9	
F10	* F6.1	F09 , X10	1	FUNCTION 10	
F11	* F6.1	F07 , X11	1	FUNCTION 11	
F12	* F6.1	F10 , X11	1	FUNCTION 12	
F13	* F6.1	F11 , F12	1	FUNCTION 13	

Keys: Esc, PgUp, PgDn, F1 = Help, F2 = Toggle, F3 = Formula, F = Update

Figure E-3. Screen 3 - Function Selection.

After reading the function file, PAYGEN will display the following for each function: function name, print flag, output format, function parameters, model type, and function description. Just like in screen 2 the user can change the output format.

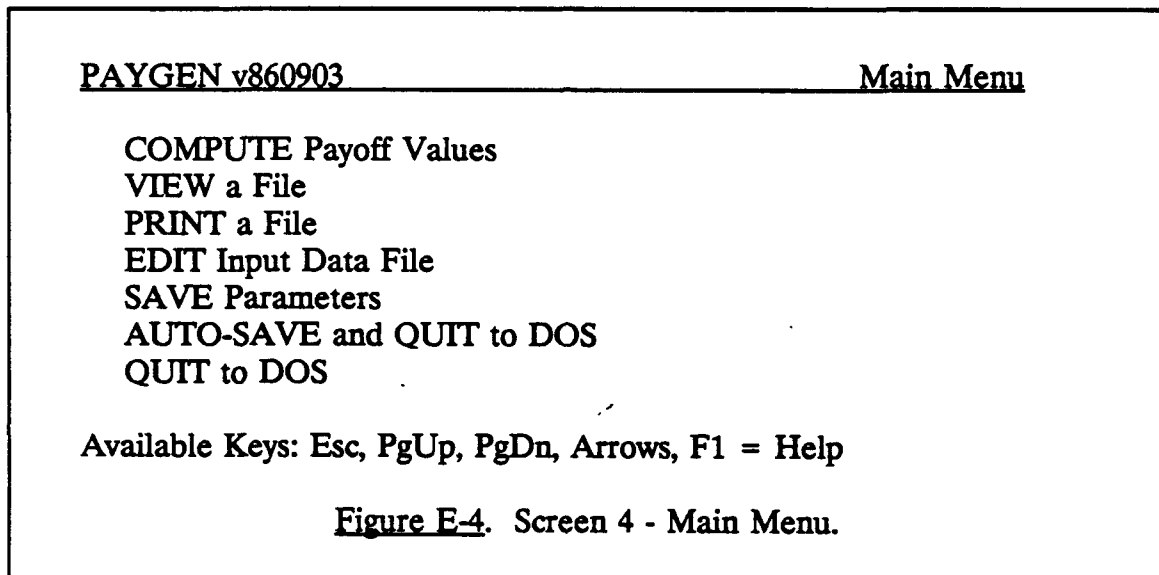
The user moves the highlight to the function to alter and types in the output format wanted. Here again the F2 key is a toggle between F0.0 and F6.1.

Another feature is provided in screen 3. The F3 key will allow the user to see the actual equation that will be used to generate the payoff value by opening a window containing the equation. To get back to the regular screen, just press any key.

## Main Menu

After all functions and variables are selected, proceed on to screen 4 by pressing the PgDn key. If for any reason the user needs to go back to any of the previous screens, just press the PgUp key.

Screen 4 is the main working screen. It contains various options to select. The first option is the COMPUTE Payoff Values option. This will cause PAYGEN to compute the payoffs for the functions selected. A report will be created in the report file name specified in screen 1. PAYGEN will also create the Output Data File containing all the selected variables and payoffs. A layout of this file will be contained in the PAYGEN report.



**Figure E-4.** Screen 4 - Main Menu.

The PAYGEN report is a multi-part report. See Appendix E-3 for an actual report.) It contains a header page, output data file layout, payoffs, summary statistics, and correlations. The header page will list the time and date the report was produced, files used, and number of records read.

The second option is the VIEW option. This option allows the user to view any ASCII type file. When this option is invoked, a small window will appear with file names in it. The file names are those from screen 1. The last one, marked other, will prompt the user to enter a file name. Move through this small menu by using the up and down arrow keys. When the file you want to view is listed, just press return. It will take a few seconds to read the file into memory. After the file is displayed, move through it by using the arrow keys, PgDn and PgUp, Home and End keys. Press Esc to leave the View module and return to the main menu.

The third option is the PRINT option. This option will cause any ASCII file to be printed on the printer with the headings from screen 1. The printed output will have top

and bottom margins of 6 lines. Like the VIEW option, a small window will appear with a list of file names. Highlight the one to print and press return. Use "OTHER" if to print something other than what is listed there.

If at any time during the print operation the user wishes to stop printing, just press the Esc key. A prompt will appear on the screen asking if the user wants to continue. A "Y" or "N" reply is all that is needed to continue or stop printing.

The fourth option is the EDIT option. This option reads the Input Data File into memory, and allows you to change data values. The editor is variable oriented rather than line oriented. The arrow keys will move from variable to variable, or record to record. On the bottom left of the screen the name of the current variable, along with its value is displayed. To change the value of the variable, simply press the F2 key. A cursor will appear in the box on the lower left portion of the screen. Enter in the new value and press return. The value will automatically be right justified in the box.

The editor determines the length of the data value fields by scanning the current record. It is assumed that there is at least one space between the data values in the Input Data File. The editor keys in on that space and uses the remaining spaces and the length of the number to determine the number of characters in the data field. This should be taken into account when creating the Input Data File. Allow enough space between the variable data values to allow it to hold the largest number. Keep your columns right justified as it makes it easier to view the data in the editor. Even though the editor doesn't require the data to be right justified into columns, it should make work easier.

Press Esc to exit the editor. If changes have been made the user will be prompted to save the changed file. A "Y" will cause the current Input Data File to be overwritten with the changed version. A "N" response will leave the file unchanged.

The fifth option is the SAVE parameter option. This option causes the parameter file to be written with the information contained in screen 1. With the parameters saved in the parameter file, there is no need to re-enter the data contained in screen 1. This is very handy if file names other than the defaults are used.

The sixth and seventh options will exit out of the PAYGEN program. The AUTO-SAVE and QUIT to DOS option will save the parameter file and exit from the program. The QUIT to DOS option will just exit from the program without saving parameter information.

### III. ERROR HANDLING

PAYGEN has a built-in error handling routine. When a field has been input incorrectly an error message will appear. The error message contains two types of information: what is not acceptable and how to correct it. A listing of the error messages can be found in Appendix E-4.

APPENDIX E-1: SAMPLE INPUT DATA FILE

80	70	90	90	80	90	40	80	2	2	50
80	70	90	90	80	90	40	80	2	9	50
80	70	90	90	80	90	40	80	10	2	50
80	70	90	90	80	90	40	40	2	2	50
80	70	90	90	80	90	10	80	2	2	50
80	70	90	90	80	60	40	80	2	2	50
80	70	90	90	40	90	40	80	2	2	50
80	70	90	20	80	90	40	80	2	2	50
80	70	20	90	80	90	40	80	2	2	50
80	100	90	90	80	90	40	80	2	2	50
20	70	90	90	80	90	40	80	2	2	50
20	100	20	20	40	60	10	40	10	9	50
20	100	20	20	40	60	10	40	10	2	50
20	100	20	20	40	60	10	40	2	9	50
20	100	20	20	40	60	10	80	10	9	50
20	100	20	20	40	60	40	40	10	9	50
20	100	20	20	40	90	10	40	10	9	50
20	100	20	20	80	60	10	40	10	9	50
20	100	20	90	40	60	10	40	10	9	50
20	100	90	20	40	60	10	40	10	9	50
20	70	20	20	40	60	10	40	10	9	50
80	100	90	20	40	60	10	40	10	9	50
80	70	20	20	80	90	10	40	10	2	50
80	70	20	20	80	90	10	40	10	9	50
80	70	20	20	40	60	40	80	2	2	50
80	70	20	20	40	60	40	80	2	9	50
80	70	20	20	40	60	40	80	10	2	50
80	70	20	20	40	60	40	80	10	9	50
80	70	20	20	40	60	10	40	2	2	50
80	70	20	20	40	60	10	40	2	9	50
80	70	20	20	40	60	10	40	10	2	50
80	70	20	20	40	60	10	40	10	9	50
20	100	90	90	80	90	40	80	2	2	50
20	100	90	90	80	90	40	80	2	9	50
20	100	90	90	80	90	40	80	10	2	50
20	100	90	90	80	90	40	80	10	9	50
20	100	90	90	80	90	10	40	2	2	50
20	100	90	90	80	90	10	40	2	9	50
20	100	90	90	80	90	10	40	10	2	50
20	100	90	90	80	90	10	40	10	9	50
20	100	90	90	40	60	40	80	2	2	50

APPENDIX E-2: SAMPLE OUTPUT DATA FILE

90,	90,	80,	90,	40,	71,	454.03,	355.258,	81
90,	90,	80,	90,	40,	18,	454.03,	88.020,	54
90,	90,	80,	90,	40,	58,	454.03,	291.053,	75
90,	90,	80,	90,	40,	64,	454.03,	317.877,	77
90,	90,	80,	90,	10,	64,	454.03,	318.690,	77
90,	90,	80,	60,	40,	71,	409.03,	355.258,	76
90,	90,	40,	90,	40,	71,	394.03,	355.258,	75
90,	20,	80,	90,	40,	71,	430.26,	355.258,	79
20,	90,	80,	90,	40,	71,	358.97,	355.258,	71
90,	90,	80,	90,	40,	71,	413.60,	355.258,	77
90,	90,	80,	90,	40,	71,	332.75,	355.258,	69
20,	20,	40,	60,	10,	10,	83.20,	47.578,	13
20,	20,	40,	60,	10,	54,	83.20,	272.380,	36
20,	20,	40,	60,	10,	15,	83.20,	75.727,	16
20,	20,	40,	60,	10,	9,	83.20,	43.463,	13
20,	20,	40,	60,	40,	9,	83.20,	43.169,	13
20,	20,	40,	90,	10,	10,	128.20,	47.578,	18
20,	20,	80,	60,	10,	10,	143.20,	47.578,	19
20,	90,	40,	60,	10,	10,	95.94,	47.578,	14
90,	20,	40,	60,	10,	10,	134.16,	47.578,	18
20,	20,	40,	60,	10,	10,	142.00,	47.578,	19
90,	20,	40,	60,	10,	10,	287.04,	47.578,	33
20,	20,	80,	90,	10,	54,	335.20,	272.380,	61
20,	20,	80,	90,	10,	10,	335.20,	47.578,	38
20,	20,	40,	60,	40,	71,	230.20,	355.258,	59
20,	20,	40,	60,	40,	18,	230.20,	88.020,	32
20,	20,	40,	60,	40,	58,	230.20,	291.053,	52
20,	20,	40,	60,	40,	11,	230.20,	56.690,	29
20,	20,	40,	60,	10,	66,	230.20,	330.067,	56
20,	20,	40,	60,	10,	15,	230.20,	75.727,	31
20,	20,	40,	60,	10,	54,	230.20,	272.380,	50
20,	20,	40,	60,	10,	10,	230.20,	47.578,	28
90,	90,	80,	90,	40,	71,	251.90,	355.258,	61
90,	90,	80,	90,	40,	18,	251.90,	88.020,	34
90,	90,	80,	90,	40,	58,	251.90,	291.053,	54
90,	90,	80,	90,	40,	11,	251.90,	56.690,	31
90,	90,	80,	90,	10,	66,	251.90,	330.067,	58
90,	90,	80,	90,	10,	15,	251.90,	75.727,	33
90,	90,	80,	90,	10,	54,	251.90,	272.380,	52
90,	90,	80,	90,	10,	10,	251.90,	47.578,	30

### APPENDIX E-3: SAMPLE REPORTS

Policy File Name ..... SAMPLE2.POL  
Variable Name File ..... SAMPLE2.NAM  
Input Data File ..... SAMPLE2.DAT  
Output Payoff File ..... SAMPLE2.OUT  
Output Report File ..... SAMPLE2.PRN  
Number of Input Records Read ..... 64  
Date ..... Aug 09 1986 06:49:34

#### Generated Payoffs Layout

Policy: SAMPLE  
Sample Title

ID	FORMAT	LOCATION	DESCRIPTION
X03	F5.0	1- 5	VARIABLE 3
X04	F5.0	7- 11	VARIABLE 4
X05	F5.0	13- 17	VARIABLE 5
X06	F5.0	19- 23	VARIABLE 6
X07	F5.0	25- 29	VARIABLE 7
F10	F5.0	31- 35	FUNCTION 10
F11	F8.2	37- 44	FUNCTION 11
F12	F8.3	46- 53	FUNCTION 12
F13	F5.0	55- 59	FUNCTION 13

## Generated Payoffs

Policy: SAMPLE  
Sample Title

Seq	X03	X04	X05	X06	X07	F10	F11	F12	F13
1	90	90	80	90	40	71	454.03	355.258	81
2	90	90	80	90	40	18	454.03	88.020	54
3	90	90	80	90	40	58	454.03	291.053	75
4	90	90	80	90	40	64	454.03	317.877	77
5	90	90	80	90	10	64	454.03	318.690	77
6	90	90	80	60	40	71	409.03	355.258	76
7	90	90	40	90	40	71	394.03	355.258	75
8	90	20	80	90	40	71	430.26	355.258	79
9	20	90	80	90	40	71	358.97	355.258	71
10	90	90	80	90	40	71	413.60	355.258	77
11	90	90	80	90	40	71	332.75	355.258	69
12	20	20	40	60	10	10	83.20	47.578	13
13	20	20	40	60	10	54	83.20	272.380	36
14	20	20	40	60	10	15	83.20	75.727	16
15	20	20	40	60	10	9	83.20	43.463	13
16	20	20	40	60	40	9	83.20	43.169	13
17	20	20	40	90	10	10	128.20	47.578	18
18	20	20	80	60	10	10	143.20	47.578	19
19	20	90	40	60	10	10	95.94	47.578	14
20	90	20	40	60	10	10	134.16	47.578	18
21	20	20	40	60	10	10	142.00	47.578	19
22	90	20	40	60	10	10	287.04	47.578	33
23	20	20	80	90	10	54	335.20	272.380	61
24	20	20	80	90	10	10	335.20	47.578	38
25	20	20	40	60	40	71	230.20	355.258	59
26	20	20	40	60	40	18	230.20	88.020	32
27	20	20	40	60	40	58	230.20	291.053	52
28	20	20	40	60	40	11	230.20	56.690	29
29	20	20	40	60	10	66	230.20	330.067	56
30	20	20	40	60	10	15	230.20	75.727	31
31	20	20	40	60	10	54	230.20	272.380	50
32	20	20	40	60	10	10	230.20	47.578	28
33	90	90	80	90	40	71	251.90	355.258	61
34	90	90	80	90	40	18	251.90	88.020	34
35	90	90	80	90	40	58	251.90	291.053	54

### Summary Statistics

Policy: SAMPLE  
Sample Title

Var ID	LOW	HIGH	MEAN	Std Dev (N)	Std Dev (N-1)
X03	20.000	90.000	50.625	34.725	35.000
X04	20.000	90.000	49.531	34.570	34.843
X05	40.000	80.000	58.125	19.912	20.069
X06	60.000	90.000	73.594	14.934	15.052
X07	10.000	40.000	24.531	14.993	15.111
F10	8.634	71.052	38.044	26.266	26.474
F11	83.200	454.025	237.630	107.191	108.038
F12	23.789	355.258	160.850	125.053	126.042
F13	12.637	80.928	39.848	20.178	20.337

### Correlations

Policy: SAMPLE  
Sample Title

Var ID	X03	X04	X05	X06	X07	F10	F11	F12	F13
X03	1.000	0.841	0.273	0.273	0.217	0.228	0.441	0.355	0.454
X04	0.841	1.000	0.303	0.303	0.248	0.263	0.396	0.385	0.449
X05	0.273	0.303	1.000	0.874	0.185	0.247	0.706	0.282	0.550
X06	0.273	0.303	0.874	1.000	0.185	0.247	0.697	0.282	0.545
X07	0.217	0.248	0.185	0.185	1.000	0.306	0.296	0.351	0.375
F10	0.228	0.263	0.247	0.247	0.306	1.000	0.418	0.906	0.784
F11	0.441	0.396	0.706	0.697	0.296	0.418	1.000	0.507	0.845
F12	0.355	0.385	0.282	0.282	0.351	0.906	0.507	1.000	0.889
F13	0.454	0.449	0.550	0.545	0.375	0.784	0.845	0.889	1.000

## APPENDIX E-4: PAYGEN ERROR MESSAGES

The following are error messages that are displayed by PAYGEN. After the error message you will find information on how to correct the problem.

### Unable to Read Variable Name File:

This error will occur when the Variable Name File that was specified in screen 1 could not be found. Check the name you have specified for the Variable Name File. If the name is correct, then make sure the file is on the default drive.

### No Function or Variables are Selected:

This error occurs if your trying to compute payoffs and the Variable or Policy file have not been read. These files are read when you go into screen 2 and 3. You shouldn't go from screen 1 directly to screen 4 if you're going to compute payoffs. To correct the problem, go into screen 2 and screen 3 and select the variables or payoffs you want.

### Output Format Overflow Occurred:

This error occurs when an output data element has overflowed the output format you specified. The output payoff file will contain "\*" for those values that cause an overflow condition. To correct the problem, view the output data file to find which variable or payoff overflowed. Then go back to screen 2 or 3 and change the output format for that item. Continued processing with an overflow condition will produce an invalid payoff data file, and report file.

### No File Name Returned:

This error occurs if you select OTHER in the View or Print option, and don't enter a file name. If no file name is entered PAYGEN cannot continue. To continue reselect the option and enter a file name.

**<file name> was not Found:**

This error occurs when the file specified could not be found. To correct the problem, reenter a file name of an existing file.

**<file name> is an Empty File:**

This error occurs if the file exists but doesn't contain any records. To correct the problem, reenter a new file name that contains data.

**No Input Data File Name Specified:**

**No Report File Name Specified:**

**No Payoff File Name Specified:**

These errors occur if you're trying to compute payoffs and these file names were blank. To correct the problem, go back to screen 1 and enter file names for the appropriate file.