1987

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

THE FEASIBILITY OF USING TAE AS THE UIL FOR THE SPACE STATION AND FOR OTHER INTERNAL NASA TASKS AND PROJECTS

Prepared By:	Esther Naomi Gill, Ed.D.
Academic Rank:	Associate Professor
University & Department:	Oakwood College Huntsville, AL 35896 Business & Information Systems Management
NASA/MSFC: Laboratory:	Information & Electronic Systems
Division:	Software & Data Management
Branch:	Systems Software
NASA Colleagues:	Robert L. Stevens and John W. Wolfsberger
Date:	August 21, 1987
Contract No.:	The University of Alabama in Huntsville NGT 01-008-021

2.0 ACKNOWLEDGEMENTS

My thanks go again to the Information and Electronic Systems Laboratory, W. C. Bradford, Director, Gabe Wallace, Assistant Lab Director; Software and Data Management Division, Jack Lucas, Chief; Systems Software Branch, Walter Mitchell, Chief; and colleagues Robert L. Stevens and John W. Wolfsberger who for the third year opened their laboratory, shared their facilities, and gave of their expertise and time which provided the necessary atmosphere and pleasant surroundings for this research effort.

Early in the summer, Cathy White shared her library of current TAE happenings, and Bob Stevens briefed me on the area of research I was to address for which I am grateful. Terry Brown made sure that the partitions were removed from the hard disk of my computer which permitted me to gain access and save and revise my findings and eliminated early frustrations of entering text and not being able to save it. To John Wolfsberger who invited Elfrieda Harris from the TAE Support Office, and co-editor of the TAE News, NASA/Goddard Space Flight Center, Greenbelt, MD 20771 to conduct a TAE workshop for the Systems Software Branch and me, in particular, on my topic --I'm grateful. Kaye Langford and Brenda Sparks aided in getting my viewgraphs together for my oral presentation--thanks ladies. Without the amenities and the cooperation of the aforementioned persons, this effort would have been futile.

Last, but not least, highest praise goes to ASEE, the designers of the Summer Faculty Fellowship Program; The University of Alabama, Huntsville, sponsors, Dr. Gerald R. Karr, University Program Co-Director; Mrs. Dina Engler, Dr. Karr's secretary, for your untiring efforts in keeping us informed of program happenings and for always being available to answer questions and offer assistance; Mrs. Ernestine Cothran, NASA/MSFC Assistant for University Relations and NASA/MSFC Program Co-Director; and NASA/Marshall Space Flight Center for cooperating and sharing in the implementation of the Summer Faculty Fellowship Program (SFFP) over the years.

THE FEASIBILITY OF USING TAE AS THE UIL FOR THE SPACE STATION AND FOR OTHER INTERNAL NASA TASKS AND PROJECTS

By
Esther Naomi Gill, Ed.D.
Associate Professor of Computer Science
Oakwood College
Huntsville, AL 35896

3.0 A B S T R A C T

This description of Transportable Applications Executive (TAE) is intended to serve as a determination of the feasibility of its use as the Space Station User Interface Language (SSUIL). TAE was developed by the Space Data and Computing Division, Space and Earth Sciences Directorate of the NASA/GSFC (Goddard Space Flight Center) and by Century Computing, Inc. in 1980.

TAE is an executive program that ties a system of application programs into a single easily operated whole & supports users' operations of programs through a consistent friendly and flexible interactive user interface. TAE also supplies the interface between the user and the various application programs in a particular computer system. It appears to be an effective user interface for infrequent as well as for expert users.

The UIL will support software development, on board and on ground operations, tests & diagnoses, and a family of languages (Ada, Assembly, BASIC, C, COBOL, FORTRAN, Pascal). Prospective users of the SSUIL are Requirement and Specification Engineers, Coding and Testing Engineers; Experimenters; Software Designers and Developers; Maintenance Engineers, and Crewpersons.

For most applications, the various elements of the system are so complex that users cannot communicate directly with the operational programs, but must manage the system through intermediate interfaces. An applications executive is a master program which unifies the many subprograms of the system.

TAE uses everyday terminology, is almost 100 per cent machine independent, is NASA owned, imposes no restrictions on the type of application programs supporting TAE, permits the prototyping of systems without writing any code, and can interface with FORTRAN, C, and Ada programming applications.

TAE is primarily a text-based user interface management system utilizing programs that acquire parameters or a command language which is not appropriate for all SS applications needing solutions. TAE is neither in use for real-time operations nor integration and tests which are primary UI functions.

Nevertheless, since there are more pros than cons for the use of TAE as the UIL for the SS and other NASA internal tasks and projects, I say, why not give it a try with nothing to lose and possibly everything to gain, as it is already a NASA owned/ maintained interfacing programming tool.

4.0 TABLE OF CONTENTS

5.0 PREFACE

5.1 OVERVIEW

This description of TAE is intended to serve as a determination of the feasibility of using TAE for the User Interface Language (UIL) of the Space Station (SS). This treatus is not an exhaustive examination of TAE's capabilities; details on TAE can be found in the Reference section of this document.

The Transportable Applications Executive (TAE) was developed by the Space Data and Computing Division, Space and Earth Sciences Directorate, of the NASA/Goddard Space Flight Center (GSFC) and by Century Computing, Inc., in 1980 to give programmers developing applications a standardized portable user interface.

The (TAE) is an executive program that ties a system of application programs into a single easily operated whole and supports users operations of programs through a consistent friendly and flexible interactive user interface. It was developed to better serve the needs of the end user, the application programmer, and the systems designer.

TAE also supplies the interface between the various application programs in a particular computer system and the user. TAE communicates with the human operator through menu and command interfaces, information displays, parameter prompting, error reporting, and on-line help.

It appears to be an effective user interface for infrequent as well as for expert users. It is highly portable (87% of code), supports system extendability and provides common interactive user interface to application programs.

5.2 BACKGROUND

The permanently manned Space Station calls for one of the most demanding arrays of software capabilities that any project to date has been required to provide. NASA, therefore, established a common software support environment that all contributing entities must use in the production of customized software; and it finally recommended that one language be designated for the production of all Space Station Application Software--specifically, Ada was recommended. MSFC has mission responsibility to produce the software to be used during the life-cycle phases of the Space Station for such elements as the habitation, laboratory, and logistics modules. Since these elements will include software packages provided by other work packages, the MSFC software manager is developing plans to integrate and test

software provided by other centers. This as well as those developed by MSFC will require MSFC developed software to be integrated into a total Space Station package at the same time software packages from other sources must be mated and integrated into the MSFC packages for subsequent verification of the total Space Station system.

The UIL will support software development, on-board and ground operations, tests and diagnostics, and a family of languages (Ada, Assembly, COBOL, FORTRAN, BASIC, Forth, Pascal, C, etc.). It should be device independent and transparent (friendly) to the user.

The prospective users of the Space Station User Interface Language (SSUIL) are described as:

Requirement/Specification Engineers
Software Designers and Developers
Coding and Testing Engineers
Experimenters
Maintenance Engineers
Managers
Crewpersons (Astronauts, Scientists, Flight and Ground Controllers, Payload Operators, and Application Customers)

The users will be utilizing the UIL to support such Space Station (SS) operations as:

- Planning and Scheduling
- Interacting with Real-time Modular/Payload Satellites
- In-Orbit Checkout, Repairing, Processing & Servicing, Controlling & Monitoring Space
- Environment Control and Life-Support Systems (ECLSS)
- Communicating with other Languages
- Detecting and Diagnosing Equipment Malfunctions
- Interacting with Configuration Management Malfunctions
- Monitoring On-Going Systems and Making Adjustments or Troubleshooting where Needed
- Examining and Updating Databases
- Monitoring and Recording Observations
- Interacting and Communicating with Ground Systems, Interpreting, Recording, and Presenting of data
- Communicating with Payloads
- Forecasting Potential Conditions and Making Contingency Analyses
- Validating Systems Performance

- Creating and Describing Graphics Displays
- Evaluating Program's Performance
- Monitoring Users' Use of UIL

On an interactive computer system, the user gives directions for the machine's operation. For most applications, the various elements of the system are so complex that users cannot communicate directly with the operational programs, but must manage the system through an intermediate interface.

An applications executive is a master program which unifies the many subprograms of the system. In general, an executive should facilitate the user's interaction with the computer by:

1. Standardizing the Users interface to application programs.
2. Shielding the user from the host operating system.
3. Providing a congenial environment.

6.0 TAE FEATURES

TAE consists of two distinct bodies of software: the TAE Monitor (TM) and the subroutine library. The monitor handles all user-computer communications and manages the execution of application programs. The subroutine library provides several packages of commonly needed services for application programs. TAE isolates application programs from both the interactive user interface and the host operating system. The isolation of programs from host-dependent services facilitates the porting of applications among TAE-based hosts.

6.1 FUNCTIONS AND COMMANDS

To get an idea of TAE as a system, keep in mind three basic concepts:

1. Management mode, which refers to the methods used to address the system's aggregate of resources and services.

2. Procs (processes or procedures) and parameters, refers to the resources and services of the system and how you select a feature and adapt it for a specific situation.

3. On-line help, refers to the various kinds of information--available interactively--to aid you, either on an individual topic or in the use of the system as a whole.

6.1.1 TAE Model

To build a model of TAE, we'll consider the three major types of activities carried out by the executive system and the components associated with each. Table 1 summarizes these relationships:

Table 1 - TAE FUNCTIONS AND COMPONENTS

FUNCTION	COMPONENT	DESCRIPTION
Managerial	Modes	Methods of addressing the operational components which will accommodate the user's individual level of experience with TAE.

Operational	Procs & Parameters	A collection of programs which perform the analytical and housekeeping activities necessary to complete required tasks.
Informational/ Instructional	Help	A means of supplying both supplemental information on a single topic and a systematic teaching tool for the new user.

6.2 MODES PROCS AND PARAMETERS

The user sees TAE as an interface having two modes: menu and command. Novice or casual users make their way through the system by means of menus, augmented by extensive on-line help. Experienced users have a powerful language for commanding the system and controlling the environment. Both menu and command users have online explanations of the system. In either mode, the user's ultimate purpose is the same: to execute the application functions of the system.

6.2.1 Modes (Managerial)

With TAE you may send directions to the computer in one of two modes:

1. Menu Mode where you select an operation from a numbered list presented on the screen.

2. Command Mode where you communicate with the system through the TAE Command Language (TCL).

The mode selected depends on the task to be performed and familiarity with the features of the system and with one's confidence in using TAE.

6.2.1.1 Using Menu Mode--In Menu Mode, TAE communicates with the user by means of four types of displays: (1) Menu, (2) Tutor, (3) Help, and (4) Messages.

In all four types, the screen shows two different kinds of material:

1. Contents: the instructions, information, or data you need to know at that time.

2. Prompt-Line Options: A set of choices to manage your task session.

(1) Menu: A TAE menu is normally a CRT display containing numbered entries (arranged vertically on the screen) and menu prompt-line options (arranged horizontally at the bottom of the screen above the prompt line). See Figure 1 for TAE menu example.

(2) Tutor: Once a user reaches a proc, he or she is "tutored" to enter the desired parameter values. In a tutor session, the user sees a screen containing a formatted list of parameter names and either their previously established values or their defaulted values. These values can be changed easily through the tutor session. With a tutor display, users may set new values or edit old values. Users may also save sets of parameter values for future restoration. Figure 2 shows a typical tutor screen.

Unless you are using a hard copy terminal, a TAE menu is normally a CRT display containing numbered entries (arranged vertically on the screen) and menu prompt-line options (arranged horizontally at the bottom of the screen near the prompt line). See Menu Illustration which follows for an example.

On most TAE systems, you log on as a menu user and are given a "root" menu and a menu question mark (?) prompt, which appears at the lower left-side of the screen. You then have two choices to make:

1. You can make a selection from the numbered entries.

2. You can give a prompt line option.

(3) Help: To be an effective interface, an executive must be self-contained. That is, it should be able to inform the user of its resources and give instructions for their use. TAE provides the user with help in five categories: a proc or a TAE command, a menu, the operation of a mode, the parameters of a proc, and a TAE message.

A typical TAE help display may have several screens, each of which contains context-specific explanatory information (usually in sentence form and appearing in the top part of the frame) and standard help options (arranged horizontally across the bottom of the screen).

(4) Messages: A message display is used to report an error or a piece of information to a user. It consists of a single

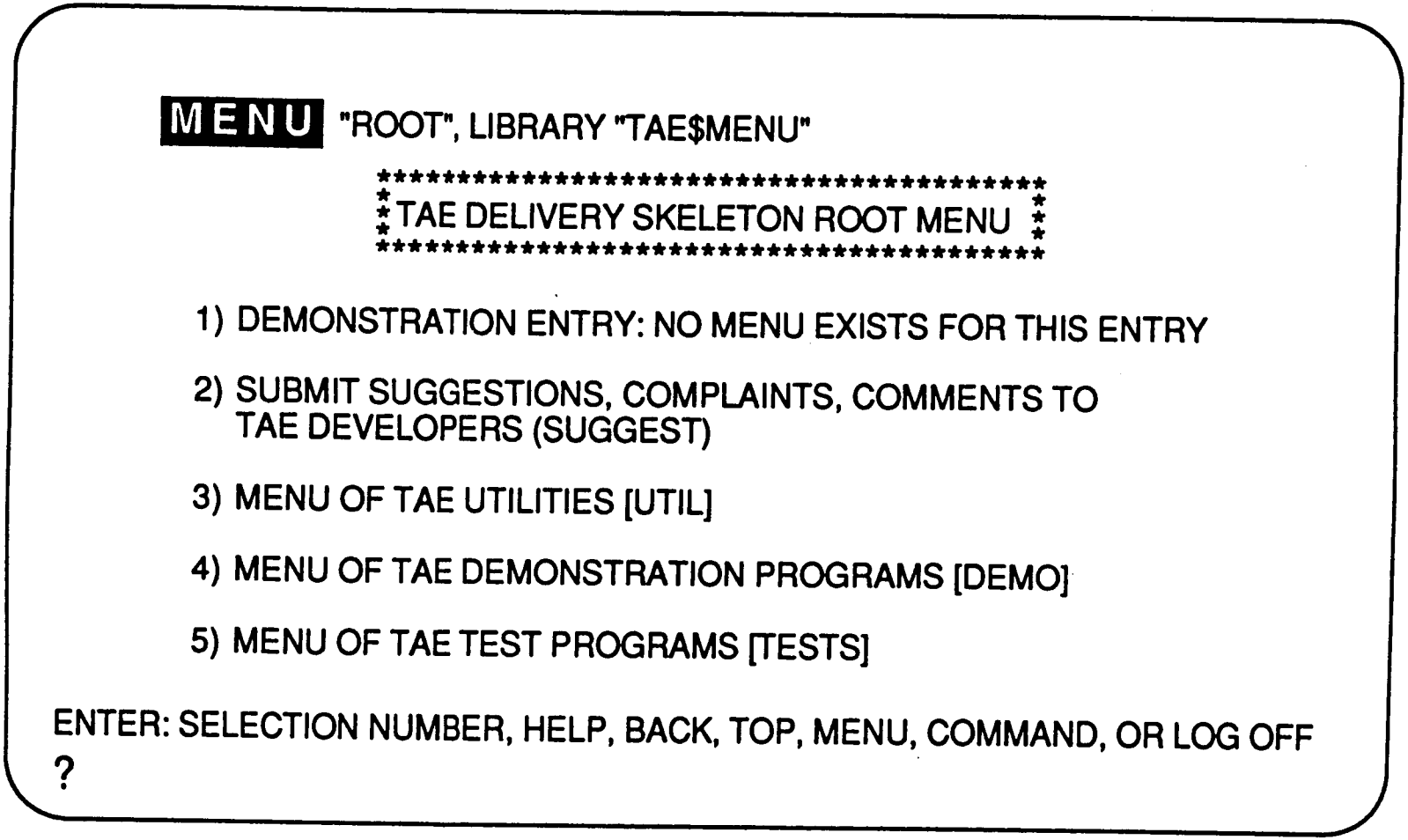

FIGURE 1

* NOTE - IF NUMBERED ENTRY NAME IS IN PARENTHESES, THE SELECTION WILL GIVE YOU A PROC. IF IT IS IN BRACKETS, IT WILL GIVE YOU ANOTHER MENU.

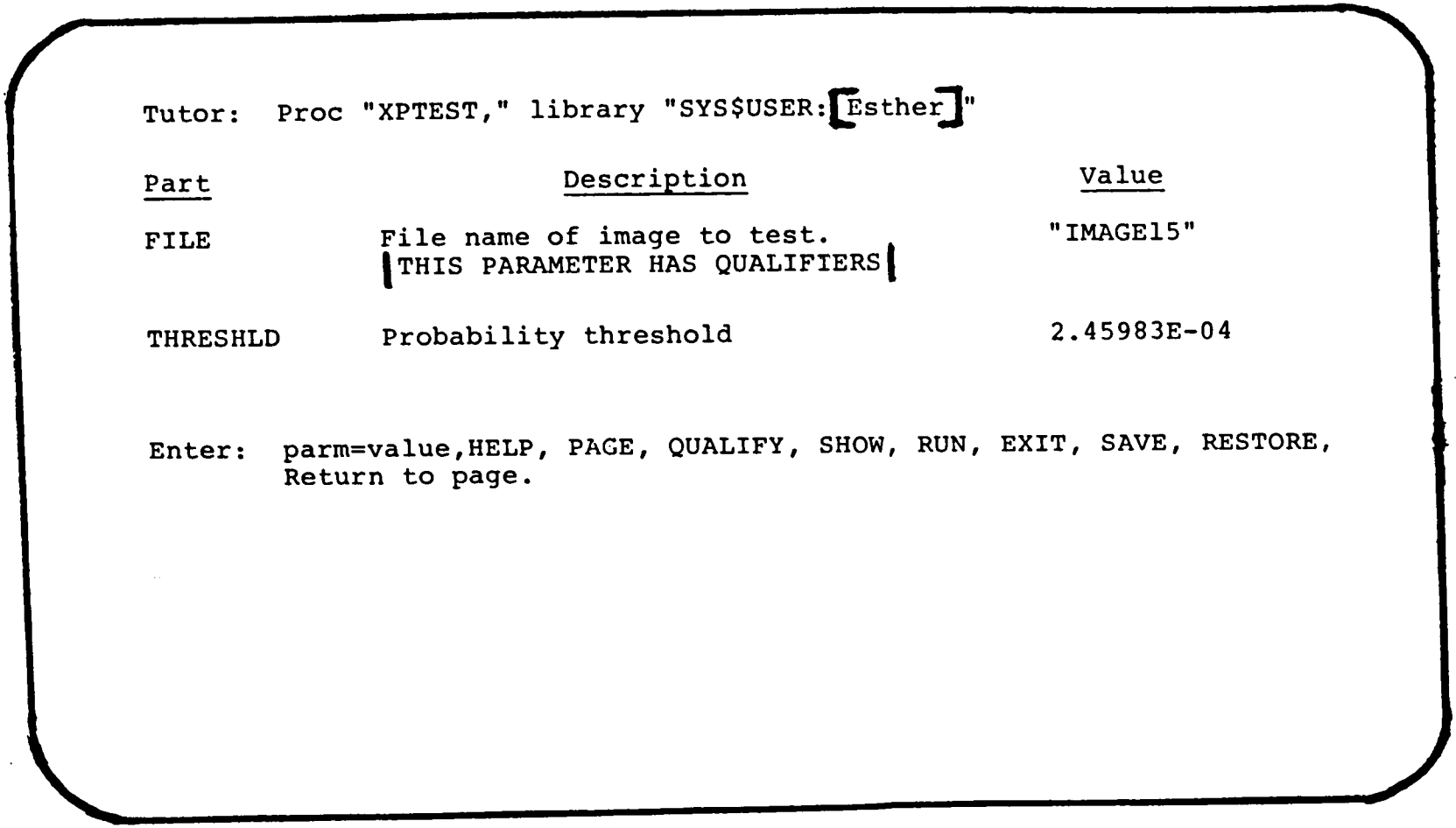

Figure 2
SAMPLE TUTOR PARAMETER DISPLAY

comment line that appears on the current display screen. The user may then choose from the options appearing on the prompt line of that particular screen. Messages may come from TAE itself, or from procs through TAE. A typical message has the form:

[WHOSE-WHAT] description

where WHOSE identifies the source of the message (e.g., TAE, ADIPS, BISHOP) and WHAT is a key (e.g., NOSUCHFILE). Any message may be supplemented by additional help, which a user accesses by typing " ? " when the message occurs.

6.2.1.2 Using Command Language Mode--Users who employ the TAE Command Language (TCL) can freely direct system activities. Unlike menu users, who have a limited set of possible actions and very little to remember at any one time, TCL users are bound only by the breadth of the system and their ability to recall how to use the various resources.

Like menu users, command users are primarily interested in executing the application and utility procs of a system. In a simple command statement, the user enters the name of a proc and parameter values, either positively or by keyboard. Command users also have access to help and messages. In addition, they may tutor on a proc or switch to menu mode at any time.

Users may also set up standard sequences of commands by creating command procedures -- collections of TAE commands, executed as a single, named function. For this purpose, the language contains control directives (e.g., LOOP, IF-THEN-ELSE) typically found in a procedural language. Other features; such as, global and local variables, substitution, assignments, input and output parameters and expression evaluation support programming through procedures.

A simple on-line command editor allows users to recall and edit or resubmit previous commands.

In TAE command mode, a user operates through a classic command line interpreter. There is a command in menu mode to enter command mode, and a command in command mode to enter menu mode.

6.2.2 Procs and Parameters (Operational)

6.2.2.1 Procs: A proc executes one of two things: (1) A requested application or (2) A utility feature. Application procs usually perform analytical tasks. Utility procs usually perform service functions.

6.2.2.2 Parameters: Once you invoke the proc you want, you may adapt it to your own use. Each proc has a parameter or set of parameters which may be given different values when the program is run for a specific reason.

Defaulted Parameters--In many cases, a parameter will have a predefined value (based on what is most convenient or most commonly used). This value is the "default;" if you do not replace it with another value, the program executes using this value. Whether you are working in Menu Mode or in Command Mode, you will have the option of entering parameter data or accepting a default value before activating the proc.

6.3 HELP (INFORMATIONAL/INSTRUCTIONAL)

Unlike some systems, help instructions for TAE will not appear of their own accord when the computer detects that the operator is making mistakes at the terminal. You must request help--on-line help is available at all times during your TAE sessions.

Help is available in several different forms on TAE. No matter what mode you are in, the system can provide you with assistance on almost any item.

6.4 TCL (TAE COMMAND LANGUAGE)

The TAE command language (TCL) is a classic command language based on Digital's DCL, the IBM System/38 command language, the UNIX shell and C shell, and the deliberations of the ANSI committee standardizing Operating System Command and Response Languages.

A TCL command line consists of the name of a proc or a TCL intrinsic command, followed by command or proc parameter values. Parameter values for any proc and most intrinsic commands are given by the user either positionally or by associating the value with the name of the parameter.

As an example, assume we have a proc named DOIT, with two integer parameters named A and B, and a string parameter named C. Assume further that the PDF for DOIT specifies that the order of the parameters is A, B, C. One may specify the initial value of the proc in any of the following ways:

```
DOIT 1, 4, hello
DOIT 1, C = hello, B = 4
DOIT C = hello, A = 1, B = 4
```

Command lines may be continued past the end of the physical terminal line; the number of continuations depends on the host facility. Comments may be appended to any command line, including continued command lines.

If the definition of a parameter specifies a default value and the user gives no value for the parameter on the command line, then the default value will be assumed. If the PDF did not specify a default value, then the TAE command line interpreter will respond with a message indicating to the user which parameters require values.

In addition to executing TAE procs, a TAE command mode user may execute a TCL intrinsic command. An intrinsic command is a built-in command, that is, a command executed by the command line interpreter. TCL provides intrinsic commands to:

- abort an executing proc;
- define an acronym for an arbitrary string of characters;
- create, delete or assign values to TCL variables;
- set the library searce order;
- start or stop session logging;
- start or stop logging all terminal I/O;
- obtain help information on a TAE intrinsic command, on a proc or proc parameters, on a menu, or on a TAE message;
- save or restore a set of TCL variables;
- initiate a tutor session;
- abort or get status on TAE async or batch jobs;
- log out of TAE.

TCL supports real, integer, and string type variables, and implicit conversion on mixed-type arithmetic operations. A value is assigned to a variable when the variable is created, or through the LET statement. TCL provides addition, subtraction, division, and multiplication arithmetic operators, a concatenation string operator, and the logical operators NOT, AND, and OR. To save the global variables to a file use the TCL SAVE command.

7.0 PROS

The advantages of using TAE to prototype the UIL are as follows:

1. Many elements of TAE's terminology are also words with eveyday meanings (such as, "HELP").

2. TAE accepts upper or lower case for user input; for example, "HELP" is the same as "help".

3. Commands and keywords may be abbreviated, even down to a single letter, so long as they are unique.

4. TAE allows for portability of itself and applications (87% of Code).

5. TAE is cost effective as it is an in-house development owned by NASA and its contractors.

6. The ability to browse TAE -- The HELP is an instructional tool and is especially valuable to intermediate learners because it allows you to request information on the topics.

7. Users of TAE on different host computers find almost identical user interfaces.

8. There are no restrictions on the type of application programs supporting TAE.

9. Programmers who write application systems to run under TAE do not have to invent their own control language or provide logic in their programs for interfacing with the user.

10. The physical appearance of TAE helps the user learn the system quickly.

11. TAE has the ability to prototype a system without writing a single line of application code.

12. TAE is able to interface with FORTRAN, C, and, most recently (see Smith Advanced Technology, Inc. Senior Systems Analysts' comments in the Reference Section of this paper) Ada programs.

13. TAE provides easily learned user-computer interface with power, flexibility, and consistency in managing application systems.

14. TAE lowers cost of software development thru savings in development efforts, reduced maintenance overhead, and decreased upgrade time.

15. TAE facilitates building a system by supplying tools for developing and customizing, interactive analysis in any discipline, and convenient implementation.

8.0 CONS

The disadvantage for using TAE for prototyping the Space Station (SS) User Interface Language (UIL) are as follows:

1. TAE graphics interface has been recently added and is not yet operationally error free.

2. TAE is primarily a text-based user interface management system utilizing programs that acquire parameters or a command language, which is not appropriate for all SS applications needing solutions.

3. Multiple users of TAE cannot communicate during shared sessions.

4. TAE does not run on a 16-bit machine at least without some function loss.

5. TAE's 87% portability claims may be overstated and therefore not good enough when one moves from machine to machine.

6. TAE is neither in use for real-time operations (some speed penalty) nor for integration and tests which are primary user interface functions.

9.0 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

Many computer activities use multiple programs to perform a task. For some applications, these programs are numerous, complex, and interrelated. Thus the user needs help in managing the various operations.

The TAE executive system provides an interface between the user and the programs in a particular computer application.

As a system, TAE contains 3 functional domains: (1) Informational/Instructional, (2) Managerial, and (3) Operational. The Informational/Instructional domain provides explanation and training on the system through the HELP component. The Managerial domain allows the user to give directions to the system through the MENU and COMMAND MODE components. The Operational domain supplies the analytical and service features of the system through the PROC and PARAMETER component.

At the least, I see TAE as a prototyping tool to be utilized by designers for the various Space Station operations as well as other internal NASA projects and tasks.

At the most, I see program designers using TAE to develop expert systems that join together various new and previously written and maintained programs which in turn will be utilized by users (of both advanced and novice types) to get answers to questions, to supply data, to control and monitor on-going systems and operations, to make adjustments where needed, to maintain, interact, and communicate with Ground Systems, to examine performances, and to detect, diagnose, and correct malfunctions, to communicate with other programming languages, etc. mentioning only some of the operations that need to be performed on the Space Station.

In order for all of these and more operations to be performed successfully, the designers will need to build user friendliness or simplicity of use into the system by keeping as many details as possible transparent to the user.

Also, since TAE can be used to interface with and develop Ada programs, it seems that this program developing tool could be included with the Ada compilers for the Micro/VAXII/VMS or UNIX disk operating systems.

I highly recommend as feasible the use of TAE as a developoment tool already owned by the Government for prototyping, program development, and interfacing with the numerous simple, compound and complex Space Station operations/tasks and with other NASA internal projects.

10.0 REFERENCES

"Application Programmer's Reference Manual for the Transportable Applications Executive Version 1.4," Century Computing Document Number 82-TAE-PGMV1H, March 31, 1987.

"C Programmer's Reference Manual for the Transportable Application Executive (Version 1.4)," Century Computing Document Number 84-TAE-CPGMV1B, March 31, 1987.

"Primer for the Transportable Application Executive (Version 1.0)" by Patricia A. Carlson, Carmen Ana Emmanuelli, Elfrieda L. Harris, and Dorothy C. Perkins, January, 1984.

Promotional Brochure TAE was developed by the Space Data & Computing Division, Space & Earth Sciences Directorate of the NASA/GSFC, Greenbelt, MD 20771, and by Century Computing, Inc., 1100 West Street, Laurel, MD 20707.

"Release Notes for the Transportable Applications Executive (Version 1.4) (VAXstation II/GFX [VMS] TAE Facelift)," Century Computing Document Number 87-TAE-GFXPLV1, February 27, 1987.

"Release Notes for the Transportable Applications Executive (Version 2.1) (UNIX Implementation)," Century Computing Document Number 84-TAE-UNIX-RELV21, June 22, 1987.

"Space Station Software Challenge to MSFC," by John W. Wolfsberger, SSE and TMIS Committee Member, August, 1987, and to be published in an ISEE upcoming issue.

"System Manager's Guide for the Transportable Appplication Executive (Version 1.4) (VAX/VMS Implementation)," Century Computing Document Number 82-TAE-SYSV1F, March 31, 1987.

"TAE News Release--August, 1987," by Albert E. Horn and Morris C. Harwell, Senior System Analysts for Smith Advanced Technology, Inc., 2009 Gallatin Street, Huntsville, AL 35802, NASA Contractors for the Space Station Operating System Study. They have recently successfully implemented and tested two methods to interface DEC VAX Ada applications with TAE. For more details, call Al or Morris at AC 205 - 533-3822 if you are a can't-wait-to-try TAE/Ada user and/or read your October, 1987, issue of TAE News.

"TAE Tutorial," Conducted July 28, and 29, 1987 at NASA/MSFC, Information & Electronic Systems Laboratory, Software and Data Management Division, Systems Software Branch by Mrs. Elfrieda L Harris, Co-editor, TAE Newsletter, published tri-annually (February, June & October)--the Newsletter contains updates on TAE

users--TAE News is published by the TAE Support Office, Image Analysis Facility, Code 635, NASA/GSFC, Greenbelt, MD 20771 and is free to users.

"User Interface Prototype (UIP) Program Plan," by John T. Ellis, TRW87-281, March, 1987.

"User-Programmer Dialogue: Guidelines for Designing Menus and Help Files for Interactive Computer Systems" by Patricia A. Carlson, NASA Technical Memorandum 84980, February, 1983.

"User's Reference Manual for the Transportable Applications Executive (Version 1.4)," Century Computing Document Number 82-TAE-USRV1H, March 31, 1987.

11.0 LIST OF ACRONYMS USED

ASEE	American Society for Engineering Education
BASIC	Beginners All Purpose Symbolic Instruction Code
C	A revision to the B Programming Language
COBOL	Common Business Oriented Language
CRT	Cathode Ray Tube
ECLSS	Environmental Control and Life Support System
FORTRAN	FORmula TRANslator or TRANslation
GSFC	Goddard Space Flight Center, Greenbelt, MD
MSFC	Marshall Space Flight Center, Huntsville, AL
NASA	National Aeronautics and Space Administration
PDF	Process or Procedure (PROC) Definition File
PROCS	TAE PROCesseS or PROCedureS
SFFP	Summer Faculty Fellowship Program
SS	Space Station
SSE	Software Support Environment
SSUIL	Space Station User Interface Language
TAE	Transportable Applications Executive
TCL	TAE Command Language
TM	TAE Monitor
TMIS	Technology Management Information System
UIL	User Interface Language