
September 1983

Volume 1, Number 1

This Is Your First Subscription Service Issue



PERSONAL COMPUTERS

OFFICE AUTOMATION

GRAPHICS

COBOL

DIBOL

Printed in the U.S.A.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	PDT
DECnet	Digital Logo	RSTS
DECsystem-10	EduSystem	RSX
DECSYSTEM-20	IAS	UNIBUS
DECUS	MASSBUS	VAX
DECwriter	PDP	VMS
		VT

UNIX is a trademark of Western Electric Corporation

Copyright © Digital Equipment Corporation 1983
All Rights Reserved

It is assumed that all articles submitted to the editor of this newsletter are with the authors' permission to publish in any DECUS publication. The articles are the responsibility of the authors and, therefore, DECUS, Digital Equipment Corporation, and the editor assume no responsibility or liability for articles or information appearing in the document. The views herein expressed are those of the authors and do not necessarily express the views of DECUS or Digital Equipment Corporation.

DIBOL/COBOL/OA/GRAPHICS/PC
COMBINED NEWSLETTER

GRAPHICS	3
COBOL.	57
OA	77
PC	122

NO SUBMISSIONS FROM THE DIBOL SIG FOR THIS NEWSLETTER

MESSAGE FROM THE COORDINATING EDITOR

Welcome to the first combined DIBOL/COBOL/OA/GRAPHICS newsletter. There have been a few surprises and I'm sure that there will be more before this beast goes to press. Obviously, schedules have been stretched a bit. Clyde Kelley of the OA SIG had planned to do the final assembly of the newsletter with tape submissions from the other three editors. However, Clyde has recently changed jobs and does not have the necessary resources for assembling a newsletter. Since I was scheduled to assemble the second newsletter, I offered to take over for this one. This all took place the last week in June, a time when most people (myself included) are preoccupied with thoughts of vacations. After very little soul searching, I decided to slide the publication schedule two weeks so I could head home to Montana for my high school reunion and a much needed backpacking trip.

But here it is, the first combined newsletter. It's somewhat ragged, perhaps lacking continuity, hardly the slick, polished magazine we would eventually like to see, but it's a start. We've got a lot to learn about putting these things together. Please bear with us.

For this first issue each SIG will have its own section, table of contents, editorials, etc. It may stay this way in the future, or it may change -- your input is heartily solicited. Please let us know what like or don't like.

Mike York
GAPSIG Newsletter Editor

FROM THE EDITOR

GRAPHICS APPLICATION SIG VOLUME II NUMBER II

FROM THE EDITOR	4
EVENT DRIVEN GRAPHICAL MENU INTERACTION	5
INTERACTIVE HIGH RESOLUTION GRAPHICS FOR THE LA-120.	13
GRAPHICS KERNAL SYSTEM	43
GAPSIG STANDARDS COORDINATOR	48
WANTED	50
ACM/SIGGRAPH	51
PRESYMPOSIA SEMINAR	53
DECUS TAPE LIBRARY	54

Pushing the deadline again. Nothing new, I've done it many times. The only problem is that I used to be a solo operation, but now there are three other SIGs involved. My tardiness will affect them. At any rate, this is the first combined newsletter of the COBOL/DIBOL/OA/GRAPHICS SIGs.

This newsletter has been the most difficult one to assemble yet. Many of the promised articles never made it into my mail box, forcing me to run around gathering appropriate submissions rather late in the game.

The last newsletter hit the streets in late April (meaning most people, including myself, received it in mid May). Since then, the Spring, 1983, DECUS Symposia was held in St. Louis. Being associated with a DECsystem-10, I found the Symposia to be very enlightening, considering the status of the 36-bit world in DEC's eyes (it's only appropriate, though, considering the other machines at our site are 14 PDP-15's, two PDP-11/40's and one 11/70).

There was something missing from the St. Louis Symposia that I couldn't quite put my finger on for the first two days, which I finally figured out on Tuesday -- there were hardly any hassles. The meeting rooms usually had room for all those wishing to attend, the food was adequate without any long lines, messages got on the message board in a timely fashion, there was usually an unoccupied telephone (if you knew where to look), and you didn't have to walk a quarter mile to find a bathroom. The only consistent hassles were waiting for street lights to change (an eternity) and waiting to get up the escalator in the convention center (never did find the stairs).

One of the more interesting and fun jobs that I've had since being involved with DECUS was working the information booth for two hours on Wednesday afternoon. It's amazing what people can lose -- lunch tickets, badges, airline tickets, and, believe it or not, divorce papers (I never did find out if the latter was ever claimed).

All in all, I felt the Symposia was the smoothest one that I've attended. The DECUS staff and Symposia Committee (and many, many others) deserve a pat on the back for a job well done.

Something new in this newsletter is the DECUS Scribe Service. DECUS paid computer science and journalism students to attend selected sessions and report on them. One graphic session report covered by the Scribe Service appears in this newsletter. More will appear in the next issues, as I didn't have time to verify the reports with the audio tapes of the sessions.

Also new in this newsletter are short write ups of new and revised DECUS Library submissions that pertain to graphics. If readers find it useful, it will be a regular feature.

The two main features in this issue are write ups of three sessions presented at

the St. Louis Symposia. The first one is the session notes from "Event-Driven Graphical Menu Interaction" presented by Ted Crane and Jon Pittman of Cornell University, which provides a description of software used to aid in the development of graphics control and menu display software. The second is "Interactive, High Resolution Graphics for the LA-120" by George Hauser of Fauske & Associates, Inc. This paper may be particularly interesting to those interested in writing their own software for displaying scientific and business graphics plots and pie charts. The last one is a report presented by the Scribe Service on the session presented by Todd Garcia of Tektronix on GKS.

One last item - submissions. Your submissions are encouraged and needed. Preferable input is 1600 bpi tapes in either BACKUP/INTERCHANGE format (for the 36-bit world) or DOS format for the rest. Although not preferred, hard copy will be accepted. Please help. The deadlines for the next two newsletters are October 15 and December 1.

EVENT DRIVEN GRAPHICAL MENU INTERACTION

Ted Crane and Jon H. Pittman
Cornell University
Ithaca, New York

PROBLEMS ENCOUNTERED WHILE DEVELOPING INTERACTIVE GRAPHICS PROGRAMS

Using current methods, application programmers must provide code to display graphics menus, interact with these menus, and organize flow of control in the program. This usually is done before the application portion of the program is written. Programming the menu results in a large expenditure of time and effort before the programmer may directly address the problem at hand.

- Application programmer is forced to deal with implementation (i.e. syntax) on control/interaction rather than with the application (i.e. semantics).
- Menu display and control structure coding is spread throughout program and even duplicated in several different places within a program.
- "Reinventing the wheel" by writing new menu display and program control code for each application program (or worse, adopting the menus and control structure of a previously developed program without understanding the reasoning behind them).
- Lack of consistent means by which interaction is implemented thereby resulting in inefficient use of the underlying graphics software.

GENERAL SOLUTIONS FOR GRAPHICS INTERACTION

- Isolate menuing and control functions from the user program.

- Provide user with a means of quickly defining and modifying menu/control structure and an easy means of integrating menu/control structure and application modules.
- Encourage (enforce?) modularity by separating application routines from program structure and allowing them to operate as independent entities.
- Provide a set of "primitive" interaction devices which handle both input and syntactic feedback functions.

PREVIOUS ATTEMPTS TO ADDRESS THESE ISSUES

- Menu design programs:

Generally produce graphics display data in form of source language. Include (or require) files which are inserted into program and compiled. May also produce actual language specific source code to perform program flow control.

These files are usually edited by the application programmer to incorporate tests and coding specific to the application. Unfortunately, this leaves a final version of the program as a mixture of graphic interaction and application code.

- Virtual Devices:

Library routines to deal with physical devices and to simulate physical devices graphically (for example, a graphic menu switch which performs as a virtual pushbutton).

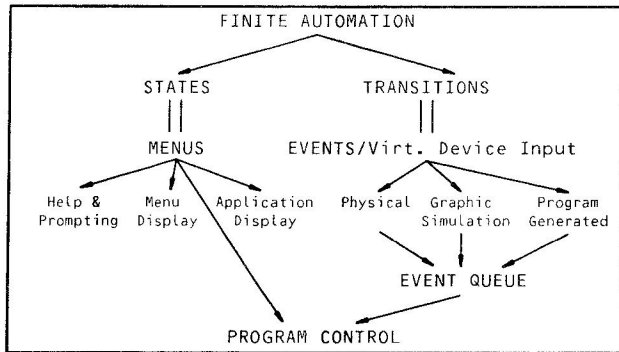
Provide the means to easily program individual instances of menu interaction but do not incorporate overall menu/control structure.

PROPOSED SOLUTION

Describe the menu interaction and program control structure as a finite state machine. Each state in the FSM is equivalent to a graphic menu. Transitions between states (menus) correspond to valid inputs received from virtual devices.

Each menu has a set of valid virtual device inputs defined for it. The interactive menu display is dependent on the set of virtual devices -- each device can a graphics representation associated with it.

Application specific routines may be called whenever a transitions from one menu to another is taken.



Components of Menu Interaction System

Finite Automata are used in many traditional programming applications, such as compiler design and text editors, but are a useful means of structuring a graphics program as well. Since graphics control flow is often repetitive code (a set of case or goto statements), the control structure can be represented as a transition diagram in the table form. A general purpose routine can be provided to direct flow of control along paths specified in the transition table. The same table can also contain information regarding the treatment and display virtual devices.

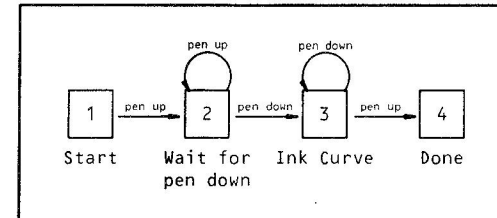
```

!
! Inking Loop, sample coding.
! Note application-specific code (underlined>
!
POINT COUNT = 0
do
begin
! display menu - may be lots of code
! read new pen value - via graphics package
end

until PENSWITCH eqLU DOWN
do
begin
POINT COUNT = POINT COUNT + 1
POINTPOINT COUNT,XJ = PEN X
POINTPOINT COUNT,YJ = PEN Y
! display existing inked line
! display menu - duplicate of above
! read new pen value - via graphics package
end
while PEN_SWITCH eqLU DOWN

```

Sample program - Inking Loop



Sample Finite State Machine - Inking Loop

Current State	Input Pen Token	
	UP	DOWN
	New State	
1 Start	2	invalid
2 Wait pen Down	2	3
3 Ink Loop	4	3
4 Done	none	none

FSM Table to Describe Inking Loop

VIRTUAL DEVICES

Most devices process user input tokens (i.e., a button push, a string read from a keyboard, etc.) and emit "events" which are placed in a queue to be processed by the run-time software.

- Keyboard - Either a physical keyboard (attached to terminal or graphics device) or a graphic simulation.

Each input token is terminated by a <CR>. When input data is read, may be checked for content (text string, numerics, evaluated expressions). If "correct" input choices are defined by menu, invalid input is rejected.

- Buttons - Toggle (on/off) and Momentary Contact Button boxes, keyboard function keys, keyboard control characters, and graphic simulations.

Every button push generates an input token. Momentary Contact buttons always generate "CLOSED" events, Toggles invert state saved from previous input and generate either "OPEN" or "CLOSED" event.

- Valulators - 1/2/.../n Dimensional Dials and slide switches (1D), joysticks and tablets (multi-D) and graphic simulations.

Generates tokens which are N-tuples of input values. Each N-tuple is treated as one event.

- Clocks.

- Repeating Global Clock - action routine gets called every "N" seconds. Useful for continuously changing displays, collecting program statistics.
- One Shot Timers - activated "N" seconds after a menu is entered (or re-entered). Used to prompt for input or timeout (and go to a different menu?).

EVENT QUEUE

The virtual devices implemented by run time software generate events and place them on an event queue. When an event is available on the input queue, the menu software dequeues it and evaluates it in the context of the current menu. Unexpected events (touching an inactive button, for example) may be rejected and, if requested by the application designer, a warning issued. When a match occurs between the event and a transition defined for the current menu, the menu software may call an action routine (so the application can make use of the call input) and the menu software may change state to a new menu. Once the event has been processed, the queue entry is deleted.

The run time software is divided into two independent portions which communicate through the event queue. The "virtual device input" portion of the software runs at an elevated level (in the VAX/VMS environment, AST driven). The "menu control" software is normally idle, waiting for an event to be placed on the event queue (in the VAX/VMS environment, hibernating or waiting for a local event flag). Sufficient memory must be allocated to the event queue so that the potential combination of a fast input device (say, a free running digitizer) and a slow transition action routine will not overflow the queue. When events are missed, a warning message is displayed.

SAMPLE MENU VOCABULARY AND EXAMPLE

MENU name = (...) Declares a menu. May be used to reference another menu (as a "submenu") when certain input functions are commonly used.

APPLICATION_DISPLAY Define an area on the screen to be used by the application program. Call an action routine to display application data.

ACTION_ROUTINE Name of an application-defined routine which is called (depending on usage) when the program begins or terminates, when menu is entered or exited, or when a transition out of menu is taken. Can also be one of several standard provided by menu software.

NEW_MENU When specified with a menu transition, the name of a new menu to be entered if that transition is taken. If not specified for a transition, the transition loops (returns to the current menu). Can also cause a return to the previous menu.

PROMPT A menu transition taken if no user input is received during a specified delay after the menu is entered. May cause prompt text to be displayed, an action routine called, and a new menu chosen.

<VIRTUAL_DEVICE> A virtual device which can provide input and menu transitions. If GRAPHIC, selects an area of the display screen and causes display of a graphic simulation of the device. Responds to user input by (optionally) calling an action routine, then (optionally) leaving the current menu and going to a new one. <VIRTUAL_DEVICE>s include KEYBOARD, VALUATOR 1D, VALUATOR 2D, BUTTON, TEXT, etc.

HELP A "standard toggle button, places menu software into "help mode", where activating any virtual device displays help text associated with that device. Default text is available for each type of device, and individual menu transition specifications can provide application specific text.

GRAPHIC_ENTITY Provides for graphic display of non-application, non-menu-related data -- program identification, programmer conceits, etc. Includes TEXT, LINE, BOX, etc.

```
! single menu state to implement
! drawing loop of sample inking program

menu INK_LOOP =

! provide for application display of inked line
(application_display=
  (<x_min>,<y_min>) ! defines area on screen
  (<x_max>,<y_max>) ! to be used by application
  DISPLAY_INK_CURVE ! and routine to display in it

! show a prompt after 10 seconds

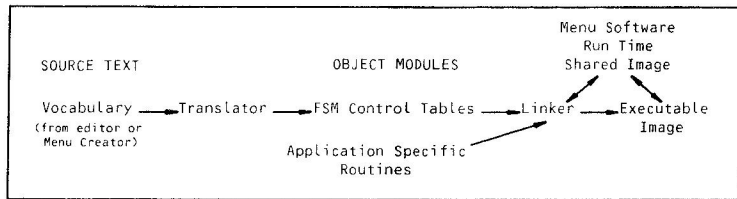
prompt=(10,"Draw curve, pick up pen when done")

! allow inking input in main viewport

valuator2d=
  (graphic <x_min>,<y_min>,<x_max>,<y_max>
   absolute range=(0.0,1.0,0.0,1.0)
   action_routine=ADD_POINT_TO_CURVE
   new_menu=INK_LOOP ! return to this menu
  )

! when pen goes up, exit to next menu
pen_up(new_menu=NEXT_MENU)
)
```

Sample Menu State for Inking



MENU SOFTWARE IMPLEMENTATION

MENU SOFTWARE IMPLEMENTATION

When implemented in a VAX/VMS environment, the menu software is divided into three major components:

- Translator/Compiler -- a standalone utility which translates the source language describing the program finite state machine into tables. These tables are subsequently interpreted by the run time system to generate graphic menu displays and control program execution. The tables are produced in the form of object modules, i.e. lists of commands and data to be processed by the VAX/VMS linker. The FSM may be described in one source file or several separate modules since individual menus are each labeled by a unique global symbol which can be resolved by the linker.
- Run Time Support -- a collection of library routines gathered together in a VAX/VMS shared library image.

Since the routines in this image are intended to control the actual program execution and VAX/VMS shared images do not contain a program transfer address, it is necessary for the main routine of the application image to call the "main" menu startup routine. This may be achieved by a small amount of coding in the application program, but it is preferable to specify one menu as the "main" menu. The translator then generates a minimal routine which calls the run time support library and emits object code instructing the linker to use the address of that routine as the linker to use the address of that routine as the program transfer point.

- Menu Creator -- an application program which can be used to interactively generate the menu vocabulary source files.

In addition, the menu software relies on the actual vocabulary for describing menus and a database of virtual devices.

STEPS REQUIRED FOR USE OF THE SYSTEM

1. Define states and transitions necessary to implement the application program. The definition may initially be small and grow during program development.
2. Write dummy routines to represent initial application code. Compile these routines.
3. Design a graphic menu, if any, to implement each state. The menu creator may be used to speed this process. Commonly used submenus may be defined and used within several major menu displays. Some menus, such as those which allow input only from physical devices (say, a keyboard), may not require graphic menus.
4. Get the menu vocabulary source files in good shape.
5. Translate description into object code (using translator).
6. Link prototype program to test menu concepts, interaction, and graphic representation. If changes are desired, return to menu or text editor.

Having done all the hard part ...

1. Replace dummy action routines with working application program.

The authors would like to thank Carlos I. Pesquera, whose "Menu Creator" program was used to draw the figures in these notes.

INTERACTIVE HIGH RESOLUTION GRAPHICS FOR THE LA-120

George M. Hauser
Fauske & Associates, Inc.
16W070 West 83rd Street
Burr Ridge, IL 60521

Fauske & Associates, Inc. uses its VAX 11/750 for scientific and engineering applications. Development of both large codes and small models as well as reduction of experimental data dictated that some form of graphical output capability be added.

Our goals in the addition of graphics capability were as follows:

1. To make the maximum use of existing equipment.
2. To keep operator involvement in the production of graphical output to a minimum.
3. To make plots of high enough resolution to see even fine detail in the models.
4. To make the package as modular and device independent as possible.
5. To make a package that is useful for both programming and non-programming personnel.
6. To do the above as inexpensively as possible.

For the most part the above goals have been met by purchasing an electronic circuit board which allowed program operation of the individual print head wires of our existing LA-120 printer/terminal, and the development, in house, of a graphics software package that can utilize it. We are able to now make use of the LA-120 as both printer and plotter. The graphical output is spooled to the print queue along with the print files so there was no increase in operator work load. Resolution on the modified LA-120 is 110 dots/inch horizontal and 72 dots/inch vertical producing output of more than adequate quality for our purposes. We have kept the package modular and device independent and could add a different plot device by adding only a single driver subroutine for that device. We have both a VAX-11 FORTRAN callable graphics library for our programmers and an interactive, menu driven front end for our non-programming personnel. Finally the cost was held low by minimizing the purchase of hardware and by using in house talent to develop the software. The purpose of this paper is to describe that software package.

The FAI Graphics package for VAX Systems has three basic parts. An interactive front end (LOOK), a device independent graphics library (PLOTLIB), and a unique device driver (SG120R) for DEC LA-120 terminals that have been modified with an available graphics board.

First, for non-programmers, an interactive menu driven front end has been developed. These programs allow non-programmers to use most of the capabilities of the plot package immediately and in a very flexible way. Any terminal can be used for interacting with this system. If a VT-100 is used, however, the preview of the plots that a user sees is slightly enhanced by using the special VT-100 graphic symbols. The front end consists of two FORTRAN-77 programs and a COM file and is initiated by the user by typing the command LOOK.

The second part is the library of FORTRAN-77 plot routines. These routines are called by one of the interactive programs to assemble the plots according to the menu items selected. They may, of course, be called directly from a users native mode program and, for programmers, this is the best way to use all of the capability built into the package. The routines in the library are "device independent". They assume ONLY that the device which will eventually draw the plot can move or draw from one point to another. No special device capabilities (ability to plot characters, draw arcs or shade areas, for example) are assumed. Therefore less sophisticated plot devices may be utilized. This means that when a new plot device appears on the scene, a single driver subroutine (PLOT(x,y,ipen)) is all that need be written in order to have it up and running. Furthermore, that subroutine can be written very easily and quickly. Plots may be sent to different devices by simply relinking the application with a different PLOT subroutine.

The third part of the package is a unique driver (PLOT) subroutine for an LA-120 terminal that has been modified with SELENAR's SG120R graphics board. This driver lets the thus modified terminal be treated like a vector graphics device that has the ability to draw or move to specified coordinates. The LA-120 may be used as a plotter by both interactive and batch jobs. It is queued as a printer, and plot files which are written by the applications are spooled to it via the DCL PRINT command. Because escape sequences cause the device to go into and out of plot mode automatically, both plot and normal print files may be intermixed in the terminal print queue. A similar driver for DEC's LA-50 personal printer has also been written.

THE INTERACTIVE FRONT END

(LOOK)

The use of an interactive program like LOOK is best documented through the use of examples. The program is menu oriented and asks fairly obvious questions, and is useable with a minimum amount of documentation.

Three example sessions with LOOK will be presented. The screens shown are what a user would see on a VT-100 type terminal. The session examples will each be accompanied by a discussion. All user input will be shown underlined so as to distinguish it from program output. In the following discussions, please refer to the accompanying screen pictures. The three example sessions follow:

Discussion of LOOK example number 1:

Example session 1 generates high resolution graphics from an external input file. The file has been generated by a users application program and its format is discussed first. The file contains information about the total number of parameters in the file, the title that will appear on the plots, the header information for those parameters, and the parameter data itself. Suppose we want to create such a file. The title that the plots are to have is 'PLANT RESPONSE'. The parameter names and their units are as follows:

Parameter	Units
TIME	HOURS
PRESS	PSIA
TEMP	DEG F
DELTA T	SEC

The format of such an input file is shown below:

```

4
'PLANT RESPONSE'
'TIME ' 'PRESS' 'TEMP ' 'T STEP'
'HOURS ' 'PSIA ' 'DEG F' 'SEC '
2.7780E-07 15.10 134.3 1.00003-03
2.5030E-02 43.28 340.5 0.4425
5.0060E-02 40.51 315.8 0.8666
7.5170E-02 40.07 309.0 0.4331
0.1003 38.06 296.2 0.5015
0.1253 37.83 290.9 0.4128
0.1503 37.10 283.8 0.2443
. . . .
. . . .
9.922 80.39 997.4 10.00
9.942 79.87 988.6 10.00
9.962 80.52 986.9 10.00
9.981 80.72 988.4 8.782
    
```

The number 4 on the first line of the file indicates that there are four parameters or variables in the file. The plot title is then given enclosed in single quotes. Then follow the four parameters names also enclosed in single quotation marks and separated by one or more spaces. Then follow the unit names for the four parameters according to the same format. Finally the parameter values in the same order as the headings with one or more spaces between them NOT enclosed in quote marks. The following limits apply to the file:

1. Parameter names and unit names must be less than or equal to 8 characters not including the quote marks.
2. The title must be less than or equal to 40 characters.
3. The maximum number of parameters in a single file is 30.
4. The maximum number of observations for each parameter is 500.

The parameter and unit names may be changed and lengthened to a total length of 40 characters (parameter name + unit name) interactively during the session if making a high resolution plot.

The first thing the system does after the user enters the command LOOK is to determine the type of terminal he is on. If it is a VT-100 an enhanced version is used that takes advantage of the VT-100's ability to generate special graphics characters. This makes the low resolution "preview plot" of the data look better than it otherwise would, but in no way affects the final high resolution output.

The system asks for the input file name (SCREEN 1) and then, after reading it, displays the title information and the parameter names with a number assigned to each of them (SCREEN 2). It then displays a Menu for the user to choose from. In this session the user chose "4" for PLOT DATA FROM SELECTED FILE. He selected an X parameter and Y parameter(s) by responding with single stroke number answers to questions the system asked. When he picked 1) AUTO SCALE the system scanned the data selected for plotting and scaled the plot preview based on the maximum and minimum values that were found. If he would have picked "2" instead, the same scaling algorithm would have been used, but the user would have provided the maximum and minimums.

A character mapped type of plot is then displayed on the screen (SCREEN 3). It stays there until the user pushes the RETURN key.

The options are then given to return to the main menu, spool the plot low resolution, or spool the plot high resolution (SCREEN 4). If the user had returned to the main menu there would be no graphical output generated corresponding to this plot preview, but he could go on and do any of the main menu items including make a another plot. If he would have chose spool low resolution, a character type of plot would be spooled for printing on any printer. This type of plot is displayed in Figure 1). No further questions would need to be asked for the low resolution plot, so after spooling it, the system would return to the main menu. But our user chose 3) SPOOL HIGH RES. resulting in the sub menu shown in the next screen (SCREEN 5).

The sub menu has 13 choices, of which our session used 5. They were, in order of use, 2, 3, 9, 12, and 1. Items 2 and 3 are similar and allow the user to change the axis scaling by changing the minimum tick value for the axis. Note that he is not allowed to change the scale multiplier. In our example for the X axis, he entered a "/" for no change.

For the Y axis (SCREEN 6), he changed the maximim from 11 to 9 by entering "1 2 9". The multiplier on the axis of E 1 (see Y axis in plot preview screen) means that the axis will go from 10 to 90 psia. If the user would have entered "10 20 90" (an error) the axis would then go from 100 to 900 psia.

Sub menu item 9 (SCREEN 7) will put a heavy frame from around the area defined by the axis pair.

Item 12 (SCREEN 8) will cause the lines representing the parameters that are plotted to be triple thick.

Each of the sub menu items (SCREEN 9) when completed returns to the sub menu again except item 1) NO CHANGES which will spool information necessary to

construct the high resolution plot and then return to the main menu.

Now a few words about the other 9 sub menu items. Item 4 displays the current title and allows the user to type in a new one. Item number 5 allows the user to change the X parameter name up to a maximum of 30 characters. Item 6 allows the same for the Y parameter(s). Items 7 and 8 allow the user to switch to a LOG type X and/or Y axis. Item 10 is similar to 9 except that in addition to the frame around the axis pair, a system of grid lines is overlaid on them. Item 11 allows the user to shade between any two of the selected parameters by entering the pair of parameter numbers. Item 13 will cause a symbol only to be plotted at the data points rather than connecting the points with lines.

After returning to the main menu (SCREEN 10) the user exits the LOOK program by entering a 6. The resulting high resolution plot from this example session is shown in Figure 2.

Discussion of LOOK example number 2:

Example session 2) constructs a high resolution plot intirely from typed in data.

Lets assume the user has a small table of data that he wants to plot. He would normally plot it by hand on graph paper, but has been encouraged to try out the LOOK program. The table is as follows:

Time Sec	P1 PSIG	P2 PSIG	P3 PSIG
0.	60.	60.	60.
.5	59.5	52.5	50.
1.	59.	44.	32.
1.5	58.	39.5	21.5
2.	57.	36.5	15.
3.	53.	31.5	7.
4.	46.5	25.5	5.
5.	39.	19.5	5.
6.	29.	13.5	5.
7.	18.5	9.	5.
7.5	14.	7.	5.
8.	10.	5.5	5.
8.5	7.	5.	5.
9.	5.5	5.	5.
9.5	5.	5.	5.

This time when asked for the input (SCREEN 11) file the user just pressed the RETURN key and was presented with a main menu (SCREEN 12) with all the items that need an input file removed. He chose number 9 "PLOT TYPED IN DATA".

He supplies title, heading and plot parameter information by answering the questions that follow (SCREEN 13), finally ending with the string "END/". The session at that point becomes very similar to session number 1) above. He selects AUTO SCALE and then is presented with a preview plot of his data (SCREEN 14).

When finished viewing it he pushes the RETURN key and chooses to create a high resolution plot (SCREEN 15).

Based on the preview plot, he decided to change the X axis scaling (SCREEN 16). But when he saw the current values of xo, xinc and xmax were already rescaled to his liking, he entered a "/" indicating no change.

He then rescaled the Y axis to start at 0, stepping by 1 to a maximum of 6 (SCREEN 17).

The user changes the y parameter names to something more meaningful than would fit in the 8 character limit of the preview plot (SCREEN 18).

He decides to place a grid on his plot (SCREEN 19) and make only plot symbols at the data points instead of joining the data points with lines (SCREEN 20).

He is finished with the plot and enters a 1 to return to the main menu (SCREEN 21). He could now go on with any of the shown menu items including SELECT AN INPUT FILE in which case, after reading in the file he would then be shown a full menu (see Example 1). But he chose to exit (SCREEN 22). The resulting high resolution plot is shown in Figure 3.

Discussion of LOOK esample number 3:

Example session 3) constructs a high resolution plot of a simple pie plot from typed in data.

After entering the LOOK command, the user pushes the RETURN when prompted for a file name as he did in example 2 (SCREEN 23).

He then enters an 8 in order to create the pie chart (SCREEN 24).

He then answers questions regarding the title, number of slices in the chart, the label for each slice, and the value for each slice (SCREEN 25). No preview of the pie graph or the bar chart is provided for.

The user exits the LOOK program (SCREEN 26). The resulting pie chart is shown in Figure 4.

ENTER INPUT FILE OR <CR> : NEW.DAT

PLANT RESPONSE
JUST A MOMENT . . .READING FILE.

SCREEN 1

PLANT RESPONSE
AVAILABLE PARAMETERS ARE . . .

1)TIME 2)PRESS 3)TEMP 4)T STEP
MENU :

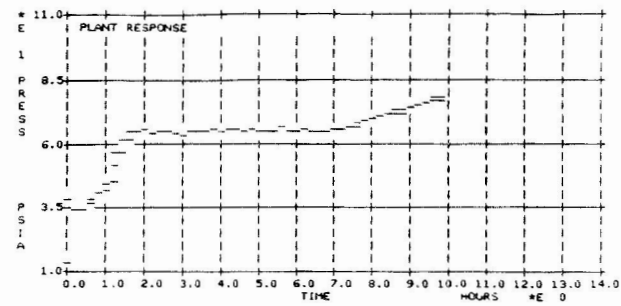
- 1) TYPE DATA FROM SELECTED FILE
- 2) PRINT DATA FROM SELECTED FILE
- 3) TYPE USER MAINTAINED FILE DIRECTORY
- 4) PLOT DATA FROM SELECTED FILE
- 5) SELECT AN INPUT FILE
- 6) EXIT
- 7) CREATE A BAR CHART FROM TYPED IN DATA
(HIGH RES. ONLY)
- 8) CREATE A PIE CHART FROM TYPED IN DATA
(HIGH RES. ONLY)
- 9) PLOT TYPED IN DATA

ENTER YOUR CHOICE : 4

SELECT X AXIS PARAMETER : 1
HOW MANY Y AXIS PARAMETERS(<=4) ? : 1
SELECT Y AXIS PARAMETER(S) : 2
1) AUTO SCALE OR 2) USR DEFINED SCALE : 1

SCREEN 2

PAGE 19



SCREEN 3

1) MENU, 2) SPOOL LOW RES., 3) SPOOL HIGH RES. : 3

SCREEN 4

PAGE 20

PICK FROM THE FOLLOWING :

- 1) NO CHANGES
- 2) CHANGE XO,XINC,XMAX
- 3) CHANGE YO,YINC,YMAX
- 4) CHANGE TITLE
- 5) CHANGE X NAME
- 6) CHANGE Y NAME(S)
- 7) CHANGE X AXIS TO LOG TYPE
- 8) CHANGE Y AXIS TO LOG TYPE
- 9) ADD FRAME TO PLOT
- 10) ADD GRID TO PLOT
- 11) SHADE BETWEEN TWO CURVES
- 12) MAKE CURVES WITH BOLD LINES
- 13) MAKE SYMBOLS ONLY

ENTER YOUR CHOICE : 2
CURRENT XO,XINC,XMAX = 0.000E+00 1.00 10.0
ENTER XO,XINC,XMAX (/ FOR NO CHANGE) : 2

SCREEN 5

PICK FROM THE FOLLOWING :

- 1) NO CHANGES
- 2) CHANGE XO,XINC,XMAX
- 3) CHANGE YO,YINC,YMAX
(FOR ABOVE 2 - DO NOT INCLUDE SCALE FACTOR)
- 4) CHANGE TITLE
- 5) CHANGE X NAME
- 6) CHANGE Y NAME(S)
- 7) CHANGE X AXIS TO LOG TYPE
- 8) CHANGE Y AXIS TO LOG TYPE
- 9) ADD FRAME TO PLOT
- 10) ADD GRID TO PLOT
- 11) SHADE BETWEEN TWO CURVES
- 12) MAKE CURVES WITH BOLD LINES
- 13) MAKE SYMBOLS ONLY

ENTER YOUR CHOICE : 3
CURRENT YO,YINC,YMAX = 1.00 2.00 11.0
ENTER YO,YINC,YMAX (/ FOR NO CHANGE) : 1 2 2

SCREEN 6

PAGE 21

PICK FROM THE FOLLOWING :

- 1) NO CHANGES
- 2) CHANGE XO,XINC,XMAX
- 3) CHANGE YO,YINC,YMAX
(FOR ABOVE 2 - DO NOT INCLUDE SCALE FACTOR)
- 4) CHANGE TITLE
- 5) CHANGE X NAME
- 6) CHANGE Y NAME(S)
- 7) CHANGE X AXIS TO LOG TYPE
- 8) CHANGE Y AXIS TO LOG TYPE
- 9) ADD FRAME TO PLOT
- 10) ADD GRID TO PLOT
- 11) SHADE BETWEEN TWO CURVES
- 12) MAKE CURVES WITH BOLD LINES
- 13) MAKE SYMBOLS ONLY

ENTER YOUR CHOICE : 2

SCREEN 7

PICK FROM THE FOLLOWING :

- 1) NO CHANGES
- 2) CHANGE XO,XINC,XMAX
- 3) CHANGE YO,YINC,YMAX
(FOR ABOVE 2 - DO NOT INCLUDE SCALE FACTOR)
- 4) CHANGE TITLE
- 5) CHANGE X NAME
- 6) CHANGE Y NAME(S)
- 7) CHANGE X AXIS TO LOG TYPE
- 8) CHANGE Y AXIS TO LOG TYPE
- 9) ADD FRAME TO PLOT
- 10) ADD GRID TO PLOT
- 11) SHADE BETWEEN TWO CURVES
- 12) MAKE CURVES WITH BOLD LINES
- 13) MAKE SYMBOLS ONLY

ENTER YOUR CHOICE : 12

SCREEN 8

PAGE 22

PICK FROM THE FOLLOWING :

- 1) NO CHANGES
- 2) CHANGE XO,XINC,XMAX
- 3) CHANGE YO,YINC,YMAX
(FOR ABOVE 2 - DO NOT INCLUDE SCALE FACTOR)
- 4) CHANGE TITLE
- 5) CHANGE X NAME
- 6) CHANGE Y NAME(S)
- 7) CHANGE X AXIS TO LOG TYPE
- 8) CHANGE Y AXIS TO LOG TYPE
- 9) ADD FRAME TO PLOT
- 10) ADD GRID TO PLOT
- 11) SHADE BETWEEN TWO CURVES
- 12) MAKE CURVES WITH BOLD LINES
- 13) MAKE SYMBOLS ONLY

ENTER YOUR CHOICE : 1

SCREEN 9

PLANT RESPONSE
AVAILABLE PARAMETERS ARE . . .

1)TIME 2)PRESS 3)TEMP 4)T STEP

MENU :

- 1) TYPE DATA FROM SELECTED FILE
- 2) PRINT DATA FROM SELECTED FILE
- 3) TYPE USER MAINTAINED FILE DIRECTORY
- 4) PLOT DATA FROM SELECTED FILE
- 5) SELECT AN INPUT FILE
- 6) EXIT
- 7) CREATE A BAR CHART FROM TYPED IN DATA
(HIGH RES. ONLY)
- 8) CREATE A PIE CHART FROM TYPED IN DATA
(HIGH RES. ONLY)
- 9) PLOT TYPED IN DATA

ENTER YOUR CHOICE : 6

SCREEN 10

PAGE 23

ENTER INPUT FILE OR <CR> :

SCREEN 11

- 5) SELECT AN INPUT FILE
 - 6) EXIT
 - 7) CREATE A BAR CHART FROM TYPED IN DATA
(HIGH RES. ONLY)
 - 8) CREATE A PIE CHART FROM TYPED IN DATA
(HIGH RES. ONLY)
 - 9) PLOT TYPED IN DATA
- ENTER YOUR CHOICE : 9

SCREEN 12

PAGE 24

ENTER TITLE FOR THIS PLOT (<=40 CHAR) : RUN NUMBER 57
ENTER NUMBER OF CURVES FOR THIS PLOT (<=4) : 3
ENTER X VARIABLE NAME (<=8 CHAR) : TIME
ENTER X UNITS (<=8 CHAR) : SEC
ENTER Y VARIABLE NAME FOR CURVE 1 : PT1
ENTER Y UNITS FOR CURVE 1 : PSIG
ENTER Y VARIABLE NAME FOR CURVE 2 : PT2
ENTER Y UNITS FOR CURVE 2 : PSIG
ENTER Y VARIABLE NAME FOR CURVE 3 : PT3
ENTER Y UNITS FOR CURVE 3 : PSIG
TO ENTER DATA FOR CURVE, ENTER THE X VALUE AND THE
Y VALUE(S) FOR THE NEXT POINT ON THE CURVE(S).
TO STOP ENTERING DATA ENTER "END/" .
ENTER X AND 3 Y VALUES : 0 60 60 60
ENTER X AND 3 Y VALUES : .5 59.5 52.5 50
ENTER X AND 3 Y VALUES : 1 59 44 32
ENTER X AND 3 Y VALUES : 1.5 58 39.5 21.5
ENTER X AND 3 Y VALUES : 2 57 36.5 15
ENTER X AND 3 Y VALUES : 3 53 31.5 7
ENTER X AND 3 Y VALUES : 4 46.5 25.5 5
ENTER X AND 3 Y VALUES : 5 39 19.5 5
ENTER X AND 3 Y VALUES : 6 29 13.5 5
ENTER X AND 3 Y VALUES : 7 18.5 9 5
ENTER X AND 3 Y VALUES : 7.5 10 5.5 5
ENTER X AND 3 Y VALUES : 8 10 5.5 5
ENTER X AND 3 Y VALUES : 8.5 7 5 5
ENTER X AND 3 Y VALUES : 9 5.5 5 5
ENTER X AND 3 Y VALUES : 9.5 5 5 5
ENTER X AND 3 Y VALUES : END/
1) AUTO SCALE OR 2) USER DEFINED SCALE : 1

SCREEN 13

PAGE 25

SCREEN 14

1) MENU 2) SPOOL LOW RES., 3) SPOOL HIGH RES. : 3

SCREEN 15

PAGE 26

PICK FROM THE FOLLOWING :

- 1) NO CHANGES
- 2) CHANGE XO,XINC,XMAX
- 3) CHANGE YO,YINC,YMAX
(FOR ABOVE 2 - DO NOT INCLUDE SCALE FACTOR)
- 4) CHANGE TITLE
- 5) CHANGE X NAME
- 6) CHANGE Y NAME(S)
- 7) CHANGE X AXIS TO LOG TYPE
- 8) CHANGE Y AXIS TO LOG TYPE
- 9) ADD FRAME TO PLOT
- 10) ADD GRID TO PLOT
- 11) SHADE BETWEEN TWO CURVES
- 12) MAKE CURVES WITH BOLD LINES
- 13) MAKE SYMBOLS ONLY

ENTER YOUR CHOICE : 2
CURRENT XO,XINC,XMAX = 0.000E+00 1.00 10.0
ENTER XO,XINC,XMAX (/ FOR NO CHANGE) : 1 2 2

SCREEN 16

PICK FROM THE FOLLOWING :

- 1) NO CHANGES
- 2) CHANGE XO,XINC,XMAX
- 3) CHANGE YO,YINC,YMAX
(FOR ABOVE 2 - DO NOT INCLUDE SCALE FACTOR)
- 4) CHANGE TITLE
- 5) CHANGE X NAME
- 6) CHANGE Y NAME(S)
- 7) CHANGE X AXIS TO LOG TYPE
- 8) CHANGE Y AXIS TO LOG TYPE
- 9) ADD FRAME TO PLOT
- 10) ADD GRID TO PLOT
- 11) SHADE BETWEEN TWO CURVES
- 12) MAKE CURVES WITH BOLD LINES
- 13) MAKE SYMBOLS ONLY

ENTER YOUR CHOICE : 3
CURRENT YO,YINC,YMAX = 0.000E+00 2.00 10.0
ENTER YO,YINC,YMAX (/ FOR NO CHANGE) : 0 1 6

SCREEN 17

PAGE 27

PICK FROM THE FOLLOWING :

- 1) NO CHANGES
- 2) CHANGE XO,XINC,XMAX
- 3) CHANGE YO,YINC,YMAX
(FOR ABOVE 2 - DO NOT INCLUDE SCALE FACTOR)
- 4) CHANGE TITLE
- 5) CHANGE X NAME
- 6) CHANGE Y NAME(S)
- 7) CHANGE X AXIS TO LOG TYPE
- 8) CHANGE Y AXIS TO LOG TYPE
- 9) ADD FRAME TO PLOT
- 10) ADD GRID TO PLOT
- 11) SHADE BETWEEN TWO CURVES
- 12) MAKE CURVES WITH BOLD LINES
- 13) MAKE SYMBOLS ONLY

ENTER YOUR CHOICE : 6
CURRENT Y NAME = PT1
ENTER Y NAME : PRESS TRANS 1
CURRENT Y NAME = PT2
ENTER Y NAME : PRESS TRANS 2
CURRENT Y NAME = PT3
ENTER Y NAME : PRESS TRANS 3

SCREEN 18

PICK FROM THE FOLLOWING :

- 1) NO CHANGES
- 2) CHANGE XO,XINC,XMAX
- 3) CHANGE YO,YINC,YMAX
(FOR ABOVE 2 - DO NOT INCLUDE SCALE FACTOR)
- 4) CHANGE TITLE
- 5) CHANGE X NAME
- 6) CHANGE Y NAME(S)
- 7) CHANGE X AXIS TO LOG TYPE
- 8) CHANGE Y AXIS TO LOG TYPE
- 9) ADD FRAME TO PLOT
- 10) ADD GRID TO PLOT
- 11) SHADE BETWEEN TWO CURVES
- 12) MAKE CURVES WITH BOLD LINES
- 13) MAKE SYMBOLS ONLY

ENTER YOUR CHOICE : 10

SCREEN 19

PAGE 28

PICK FROM THE FOLLOWING :

- 1) NO CHANGES
- 2) CHANGE XO,XINC,XMAX
- 3) CHANGE YO,YINC,YMAX
- 4) CHANGE TITLE
- 5) CHANGE X NAME
- 6) CHANGE Y NAME(S)
- 7) CHANGE X AXIS TO LOG TYPE
- 8) CHANGE Y AXIS TO LOG TYPE
- 9) ADD FRAME TO PLOT
- 10) ADD GRID TO PLOT
- 11) SHADE BETWEEN TWO CURVES
- 12) MAKE CURVES WITH BOLD LINES
- 13) MAKE SYMBOLS ONLY

ENTER YOUR CHOICE : 13

SCREEN 20

PICK FROM THE FOLLOWING :

- 1) NO CHANGES
- 2) CHANGE XO,XINC,XMAX
- 3) CHANGE YO,YINC,YMAX
(FOR ABOVE 2 - DO NOT INCLUDE SCALE FACTOR)
- 4) CHANGE TITLE
- 5) CHANGE X NAME
- 6) CHANGE Y NAME(S)
- 7) CHANGE X AXIS TO LOG TYPE
- 8) CHANGE Y AXIS TO LOG TYPE
- 9) ADD FRAME TO PLOT
- 10) ADD GRID TO PLOT
- 11) SHADE BETWEEN TWO CURVES
- 12) MAKE CURVES WITH BOLD LINES
- 13) MAKE SYMBOLS ONLY

ENTER YOUR CHOICE : 1

SCREEN 21

PAGE 29

RUN NUMBER 57

MENU :

- 5) SELECT AN INPUT FILE
- 6) EXIT
- 7) CREATE A BAR CHART FROM TYPED IN DATA
- 8) CREATE A PIE CHART FROM TYPED IN DATA
- 9) PLOT TYPED IN DATA

ENTER YOUR CHOICE : 6

SCREEN 22

ENTER INPUT FILE OR <CR> :

SCREEN 23

PAGE 30

MENU :

5) SELECT AN INPUT FILE
6) EXIT
7) CREATE A BAR CHART FROM TYPED IN DATA
8) CREATE A PIE CHART FROM TYPED IN DATA
9) PLOT TYPED IN DATA
ENTER YOUR CHOICE : 8

SCREEN 24

ENTER TITLE FOR CHART : HOME SALES FOR MAY
ENTER NUMBER OF SLICES IN PIE (<=10) : 7
ENTER LABEL FOR SLICE 1 (CHAR <=20) : MARV SWORT
ENTER VALUE FOR SLICE 1 : 4
ENTER LABEL FOR SLICE 2 (CHAR <=20) : BILL SAMUALSON
ENTER VALUE FOR SLICE 2 : 2
ENTER LABEL FOR SLICE 3 (CHAR <=20) : JERRY CRAWFORD
ENTER VALUE FOR SLICE 3 : 3
ENTER LABEL FOR SLICE 4 (CHAR <=20) : MARY ORLAND
ENTER VALUE FOR SLICE 4 : 5
ENTER LABEL FOR SLICE 5 (CHAR <=20) : SAM STEVENS
ENTER VALUE FOR SLICE 5 : 1
ENTER LABEL FOR SLICE 6 (CHAR <=20) : STEVE RAYBURG
ENTER VALUE FOR SLICE 6 : 3
ENTER LABEL FOR SLICE 7 (CHAR <=20) : ELLI JUNG
ENTER VALUE FOR SLICE 7 : 4

SCREEN 25

PAGE 31

MENU :

5) SELECT AN INPUT FILE
6) EXIT
7) CREATE A BAR CHART FROM TYPED IN DATA
8) CREATE A PIE CHART FROM TYPED IN DATA
9) PLOT TYPED IN DATA
ENTER YOUR CHOICE : 6

SCREEN 26

THE PLOT LIBRARY ROUTINES

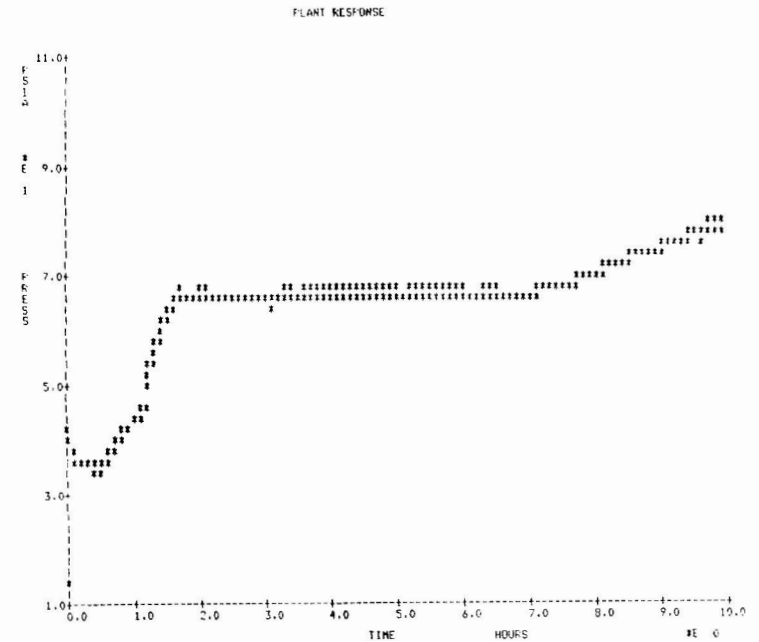


FIGURE 1

PAGE 32

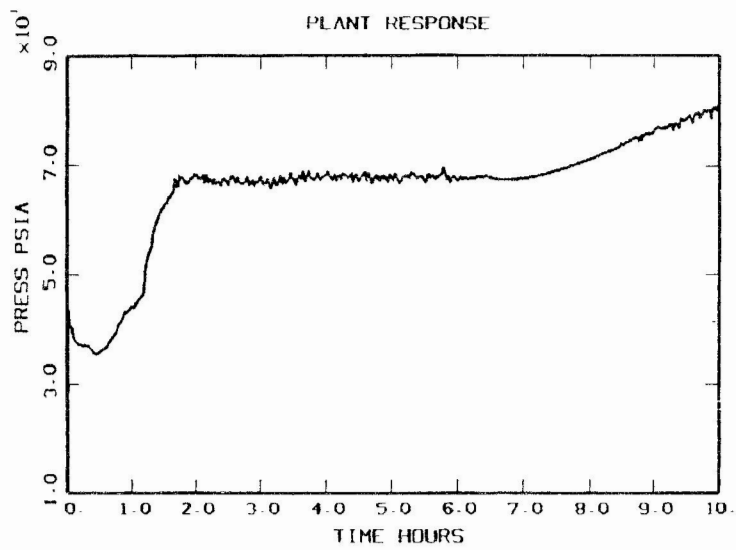


FIGURE 2

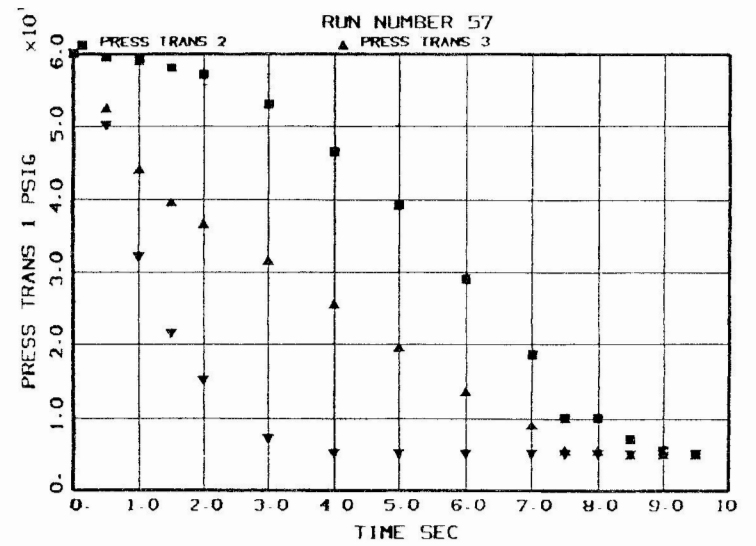


FIGURE 3

HOME SALES FOR MAY

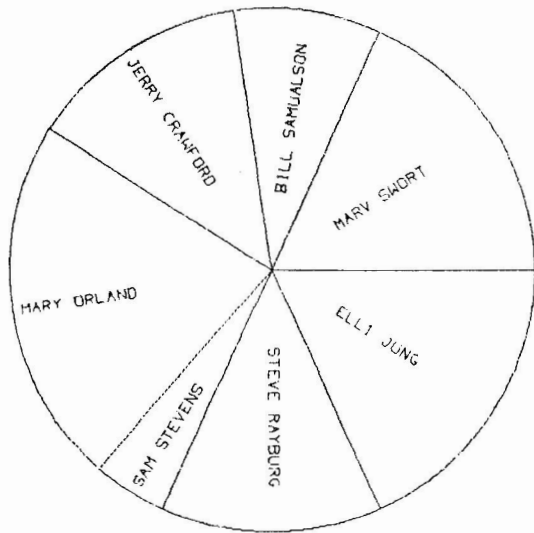


FIGURE 4

THE PLOT LIBRARY ROUTINES

The FAI plot library is designed to be used from the user's FORTRAN program. The routines exist in PLOTLIB.OLB. In order to use any of them, just include "PLOTLIB/LIB" when linking. The routines in this library are device independent and must be further linked to one of the device drivers (SG120R.OBJ of LA50.OBJ). SG120R is the driver for the modified LA-120. LA50 is the driver for DEC's LA50 printer.

The routines in the library may be divided into four categories. These are:

1. Plot Initialization
2. Plot Scaling
3. General Use
4. Plot Finalization

In general the routines should be used in this same order (1,2,3,4). The names of the subroutines and their uses are given below.

Plot Initialization Routines

OUTU Define output unit for graphics file.

PAGE Must be used initially and between plot pages to erase the plot page, and to set limits of the page size. Maximum page size is 10 inches horizontal by 8 inches vertical.

Plot Scaling Routines

XLIN Define scale values and axis length in inches for a linear axis.

YLIN Define scale values and axis length in inches for a linear Y axis.

XLOG Define scale values and axis length in inches for a log X axis.

YLOG Define scale values and axis length in inches for a log Y axis.

XAXIS Defines an axis label, a scaling factor, and the starting point for the X axis and draws the axis.

YAXIS Defines an axis label, a scaling factor, and the starting point for the Y axis and draws the axis.

FRAME Draws a heavy frame around the area defined by the axis pair and places appropriate tick marks on it.

GRID Draws a heavy frame around the area defined by the axis pair and overlays an appropriate grid on the plot.

General Plotting Routines

PLOT Used to either move or draw from the current position on the page inches to the new position on the page in inches. Can be used without the axis scaling routines since no axis scaling is necessary. The plot subroutine is actually in S6120R.OBJ.

THICK Same as plot (above) except that the drawn lines are triple thick.

PLOTS Used to either move or draw from the current position on the plot in scale units to the new position in scale units. In addition to QUTU and PAGE, if the axis scaling and drawing routines are not called first, then PLOTS has the same effect as plot.

THICKS Same as PLOTS (above) except that the drawn lines are triple thick.

TEXT Used to plot a desired text string at a desired position in inches on the page at a desired angle with respect to the horizontal using characters that are a specified height in inches.

REALNO Used to place a desired number at a desired position in inches on the page at a desired angle with respect to the horizontal using characters that are a specified height in inches.

STYLE Used to choose the style of text to be written by any of the routines that write text. There are two styles of text - 'DEFAULT' and 'CALLIG'.

SCALEX Given an X value in scale units, this FUNCTION routine returns an X value in inches on the page.

SCALEY Same SCALEX (above) but for Y values.

XUNITS Given an X value in inches, this function routine returns an X value in scale units on the plot.

YUNITS Same as XUNITS (above) except for Y values.

SIZE Used to set the Character height in inches for the axis drawing routines. If SIZE is not used, the axis routines use characters that are .2 inches high.

SYMBOL Used to place a specified plot symbol at a position on the plot in scale units.

CURVE Used to draw a curve in scale units where the X and Y values are previously stored in two arrays.

SHADE Shades in or cross hatches a polygon of any number of sides. The positions of the points in the polygon are stored in two arrays - one for X and one for Y. The polygon is closed by connecting the last point to the first. The user specifies the angle with respect to horizontal of the cross hatch lines and the distance between them in inches.

LENF Returns the length in characters of a character string. The length in inches of the string when plotted may be calculated by multiplying the number of characters by .75 times the plotted height.

SLFSCALE Returns values to be input to the axis scaling routines based on a minimum and a maximum value which is input.

FMM Used to return the maximum and the minimum values of the numbers stored in an array. Can be called repeatedly in order to find the maximum and minimum values over a series of arrays. The results usually are passed to the SLFSCALE routine (above).

ARC Draws a circular arc between two points given the two points, the radius of the arc and the center point of the arc. Optionally saves up to 500 generated points for future use by another routine such as SHADE.

Plot Finalization Routine

FIN Must be called after each plot page is complete including the last one. Signals that the bit mapped plot page should be spooled to the output file for later plotting.

FORTRAN Examples

The following programs are examples of the use of some of the PLOTLIB routines.

```

C
C   EXAMPLE PROGRAM 1
C   USES BOTH LINEAR AND LOG AXIS ROUTINES
C
C   CALL OUTU(6)           ! DEFINE THE PLOT OUTPUT
C                           ! FILE AS UNIT 6
C
C   CALL PAGE(10.,9.)
C
C   CALL STYLE('CALLIG')
C
C   CALL YLIN(-1.5, .5, 1.5, 5.)
C   CALL XLOG(.01, 2, 8.)
C
C   CALL YAXIS('Y AXIS IN LINEAR', 1.5, 1.5, 0.) ! DRAW Y AXIS
C   CALL XAXIS('X AXIS IS LOG', 1.5, 1.5, 0.)   ! DRAW X AXIS
C
C   CALL GRID              ! DRAW GRID ON AXIS'
C   IPEN=0                 ! SET FOR INITIAL DARK MOVE
C   DO 10 J=1,100
C
C   X=FLOAT(J)/100.        ! CALCULATE X
C   Y=EXP(-.3*X)*SIN(X*8.*3.14159) ! AND Y POINT
C
C   CALL THICKS(X,Y,IPEN) ! PLOT X,Y DATA IN SCALE
C                           WITH THICK LINE
C   IF (IPEN.EQ.0) IPEN=1 ! SET FOR LIGHT MOVE AFTER
C                           FIRST POINT
C
C 10 CONTINUE
C   CALL FIN               ! SPOOL PLOT PAGE TO FILE
C   STOP
C   END

```

Commands Used to Run Example 1

The following commands were given to compile, link, run and output the preceding example. The Fortran file name was EXAMP.FOR. The terminal print queue that has the modified LA-120 as an output device is TTA1:

```

$ FORT EXAMP
$ LINK EXAMP,SG120R,PLOTLIB/LIB
$ ASS/USER EXAMP.PLT FOR006
$ RUN EXAMP
FORTRAN STOP
$ PRINT/NOHEADER/QUEUE=TTA1: EXAMP.PLT

```

Figure 5 is the resulting plot from the preceding example program.

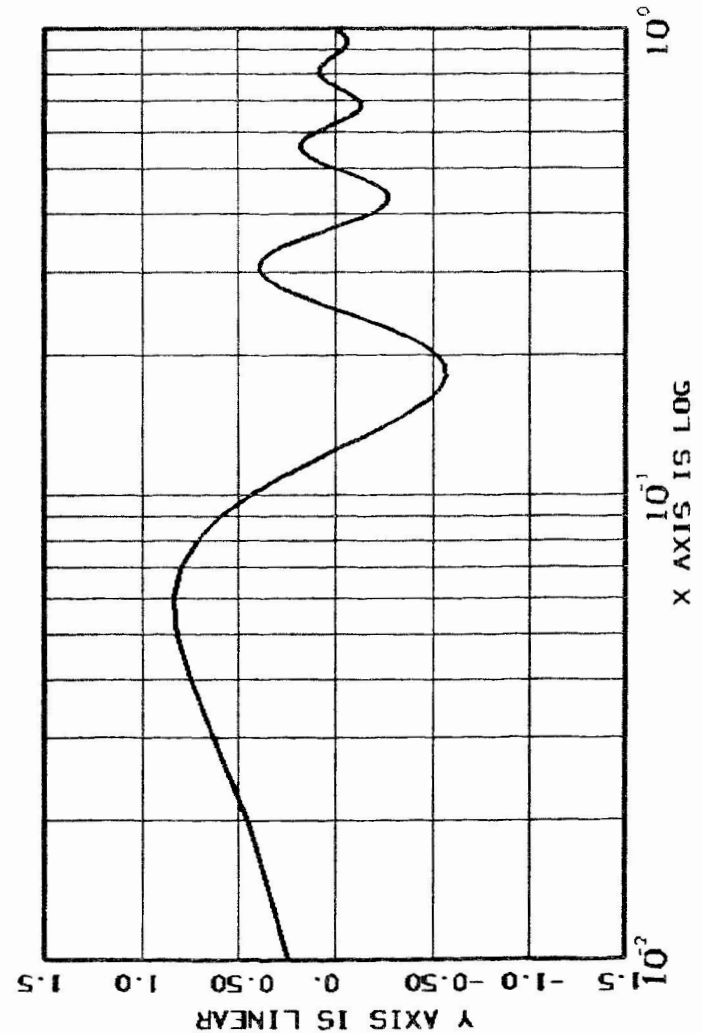


FIGURE 5

```

DIMENSION PLOX(13),POLY(13)
DATA POLX /3.,3.7,6.,3.,5.,6.,3.9,5.,5.,4.,3.8,3.,3.4/
DATA POLY /7.,4.5,3.5,2.,2.,5.1,6.7,6.1,7.,5.,5.3,4.8,6.2/

CALL OUTU(6)                ! OUTPUT PLOT TO UNIT 6
CALL PAGE(10.,8.)          ! SET PAGE SIZE & CLEAR
IPEN=0                      ! SET FOR INITIAL DARK
DO 10 J=1,13
CALL PLOT(PLOX(J),POLY(J),IPEN)    ! DRAW BOUNDARY OF POLYGON
IPEN=1
10 CONTINUE
CALL PLOT(POLX(1),POLY(1),1)      ! CLOSE POLYGON BOUNDARY
CALL SHADE(POLX,POLY,45.,-1,13)  ! SHADE IN POLYGON
CALL FIN                        ! SPOOL PLOT PAGE TO FILE
STOP
END

```

Commands Used to Run Example 2

The following commands were given to compile, link, run and output the preceding example. The FORTRAN file name was SHADER.FOR. The terminal print queue that has the modified LA-120 as an output device is TTA1:

```
$ FORT SHADER $ LINK SHADER,SG120R,PLOTLIB/LIB $ ASS/USER SHADER.PLT FOR006 $
RUN SHADER FORTRAN STOP $ PRINT/NOHEADER/QUEUE=TTA1: SHADER.PLT
```

Figure 6 is the resulting plot from the preceding example program.

SECTION III THE LA-120 PLOT DRIVER

The plot driver routine is a VAX-11 FORTRAN subroutine named PLOT which is in the file SG120R.OBJ. The calling sequence is:

```
CALL PLOT(X,Y,IPEN)
```

The parameters X and Y represent a page position in inches. The parameter IPEN is a pen up or down flag. If IPEN=0 when PLOT is called the plotting device will move to the new X,Y coordinates from its present position without drawing a line. If IPEN=1 the same move occurs, but a line is drawn along the way.

Obviously, the LA-120, even with the extra circuit installed, cannot physically act like a pen plotter. Therefore, the PLOT subroutine creates a bit map of the plot page where 0 bits correspond to no wire strike and 1 bits correspond to a wire strike or dot. When a call comes in with the new X,Y coordinate and IPEN=1, the program calculates what bits to turn on in the map to create a line to that location.

When the plot is finished, the routine creates the output plot file by writing escape sequences that will place the LA-120 in plot mode and then scans the bit map page from the top down calculating what ASCII character sequences to send so that the mapped picture is printed out. Finally the escape sequence to place the LA-120 back in character mode is sent.

The limit on page size to the size of the bit map is 10 inches horizontal by 8 inches vertical.

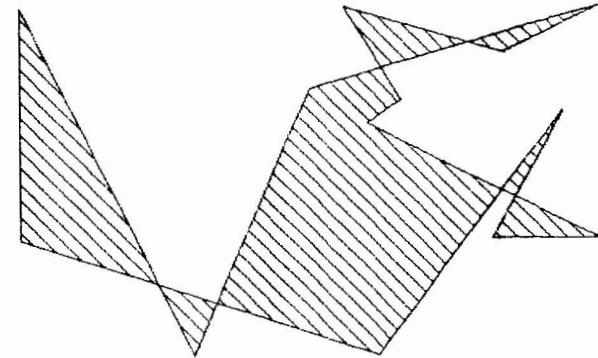


FIGURE 6

INTRODUCTION TO GKS (GRAPHICAL KERNEL SYSTEM)

Todd Garcia
Tektronix Inc.
Beaverton, OR

Jim Flatten, Session Chairperson
Ames Laboratories
Ames, IA

Reported by Gene Mitchell, DECUS Scribe Service

"GKS is very close to becoming the worldwide standard for graphics, and as such it is expected to dramatically affect the graphics industry."

So spoke Todd Garcia of Tektronix as he addressed a session entitled "Introduction to GKS (Graphical Kernel System)." The session was broken into two sections. The first was a condensed history of graphics standards; the second, a summary of current features of GKS. Tasks addressed include those of a graphics package, controlling a set of devices, picture generation, structuring of a picture, allowing graphical input, and long-term storage.

A brief history of graphics standards:

According to Garcia, "The early 1970s saw the birth of a movement for graphics standards, and it's my guess that the motivation for that movement was monetary. Since the applications were built for specific devices, the concept of device drivers was new, and the new devices coming out were cheaper and more powerful. It was in the best interests of everyone who was doing graphics to move towards them." Conversions between new and existing devices were difficult.

Some important dates were listed:

- 1976: Seillac, France Graphics Workshop: This was a meeting designed to develop new underlying concepts in graphics.
- 1977: The International Standards Organization (ISO) created a graphics working group. "Their task was to investigate current graphics packages. The result was the following quote: "No existing graphics package is suitable as a graphics standard." Also in 1977, Associated Computer Machinery (ACM), along with SIGgraph, did a report on CORE. Later, ISO recommended that the Americans (CORE) and the Germans (GKS) join their efforts to develop a common standard.
- 1978: Norway decided to use IDIGS as a Norwegian standard. The ISO working group established an editorial board. Its task was to find methods of consolidating CORE, GKS and IDIGS.
- 1979: CORE was recommended to ANSI (An American standards group) for review. In October, the ISO editorial board concluded that GKS was the most technically refined graphics package. It was presented to the working group as a draft proposal for an International standard. In December ANSI adopted CORE as a starting point for an American standard. CORE was

already a de facto standard in the U.S.

- 1980: ISO reviewed three different versions of GKS. Many issues were cleared up by the third version.
- 1981: Final technical review of GKS. All major issues were resolved, and GKS is recommended as a draft proposal.
- 1982: In June, ANSI votes against GKS because the attribute model for primitives is inconsistent with what of CORE. Later modifications to GKS led ANSI to reverse their position. In December, GKS 7.2 became a draft international standard.
- 1983: Today, CORE and GKS are both useful. GKS is only two-dimensional, while CORE is three-dimensional. GKS is transportable among application programs. CORE does not transport easily, but is established and well-known.

Garcia listed some tasks of a graphics package:

1. It must control a set of devices.
2. It must be able to generate a picture.
3. It must allow the structuring of the picture into independent proportions that can be manipulated.
4. It must allow graphical input.
5. It must have the ability to store and replay pictures.

The task of controlling a set of devices is made difficult by the many different devices available. GKS addresses the problem with the concept of the workstation. The workstation is a logical description of a group of physical devices. There are six categories of workstations in GKS:

1. OUTPUT--has, at the most, one display surface.
2. INPUT--has at least one realization of every input type.
3. OUTIN--a combination of the OUTPUT and INPUT station.
4. WISS--Workstation Independent Segment Storage. Independent of all other workstations.
5. MO--Metafile output. Long-term storage of pictures.
6. MI--Metafile input.

In GKS, a terminal itself can be a workstation because it has one display surface. Anything with more than one display surface does not qualify as a workstation. The concept behind workstations allows the application program to shield itself from the idiosyncrasies of physical devices. Every workstation in

a given category has the same capabilities. GKS allows the application program to use multiple workstations. Garcia listed four workstation subroutines:

1. Open--Opens a workstation; requires a description of the type of workstation.
2. Close--destroys a workstation and erases all information.
3. Activate--Needed for some functions that cannot be performed only with an Open workstation, e.g. generation of primitives.
4. Deactivate--Allows the application program to discriminate between workstations. Activation and deactivation can be used as often as the user chooses.

GKS uses six graphics primitives to generate a picture. They include:

1. Polyline -- one move and a series of draws.
2. Polymarker -- a series of markers.
3. Text
4. Fill area -- used to generate filled polygons.
5. Cell array -- pixel related primitive.
6. GDP -- Generalized Drawing Primitive. Essentially an escape primitive.

GKS primitives do not use a current position and are entirely self-contained (there is no such thing as a move operation). In GKS, attributes control the appearance of primitives. Of course, problems can arise when the user tries to send the same primitives to two different devices. GKS has come up with the concept of bundled attributes. For every primitive there is a set of bundled, numbered attributes. Each bundle is entirely workstation-dependent.

A criticism of GKS was that it had no individual attributes. A new hybrid version includes that feature. Garcia then listed some attributes for some of the primitives:

1. Polyline -- line type, line width and line color.
2. Poly marker -- marker type, size and color.
3. Text -- font and precision, in GKS, are tightly tied together. Support must be for both or neither. Other attributes include text path, text alignment, text color, character height, character expansion factor, character spacing and character up vector. All attributes in GKS are based on character height; once character height has been set, everything else functions as a ratio of that height to another area.
4. Fill area -- interior style, style index, and color index.

5. Cell array and GDP have no attributes.

Next, the structuring of a picture was addressed. It is structured into segments. The following segment operations are possible: Create a segment - every primitive generated will be associated with the segment; delete a segment; rename a segment. If the workstation has type widths, information can be transferred to a workstation that has been opened. Two methods are available for this: (a)Copy - which gets displayed on the surface and is a non-retained segment for that workstation, and (b)Associate - it becomes a permanent part of that workstation's segment storage.

Segment attributes include:

1. Visibility
2. Highlighting.
3. Priority.
4. Detectability.
5. Transformation.

GKS also contains six logical input devices:

1. Locator--returns a position on the display surface to the application program.
2. Pick - -returns the segment number of a primitive that was "picked" on display surface.
3. Valuator -- returns a real number.
4. Choice -- returns an integer.
5. String -- best suited for a keyboard.
6. Stroke -- returns a series of X-Y positions.

Any one of these input devices can be in one of three modes:

1. Request Mode -- the application program must ask for input and wait.
2. Sample -- the application program will test the current value of something.
3. Event -- means of receiving asynchronous input.

These devices may make it difficult to implement the hardware for them. The logical description can be a combination of hardware, firmware and software. The same is true for input devices. Some methods of physical realization include:

1. Trackball can be used for Locator.
2. A dial that generates a series of numbers or integers can be used for Valuator.
3. Two dials can be used as a Locator.
4. A series of buttons, each associated with a different integer, can be used as a Choice.
5. A Tablet can be used as a Pick operation.
6. A keyboard can be used as a valuator.

A graphics terminal with a keyboard can be an input device, an output device, and an out-in workstation.

There are two special workstations for long-term storage. The first is an output metafile. This, when open, receives primitives, attributes, bundles, etc. The second is the metafile input.

In response to a question, Garcia noted that work is now underway to give GKS 3-dimensional capability, though it is not presently available.

GAPSIG STANDARDS COORDINATOR

My name is Jim Flatten. I am an Associate Computer Scientist at Ames Laboratory, a DOE laboratory in Ames, Iowa. I work in the Computer Services Group. Our primary mission is to produce data acquisition and control software for physics and chemistry experiments. Most of our systems are RT-11 based.

In January of 1982 the laboratory installed two VAX 11/780's and about the same time I was asked to serve on the laboratory's computer graphics committee. This committee has had considerable influence on the graphics software and hardware purchased by the laboratory. Our first graphics package was specified to be compatible with the SIGGraph CORE system. This was my introduction to graphics standards.

Since that time I have been involved with the installation, maintenance and planning for graphics systems at the laboratory. I accepted the position of SIG standards coordinator at the Fall, 1982, DECUS Symposia.

The GAPSIG standards effort received a strong push forward at the St. Louis Symposia when DECUS funded ANSI participation by the SIG. I will serve as the SIG representative to the ANSI X3H3 computer graphics technical committee. The purpose of this participation is to provide a two way exchange of information between the GAPSIG members and ANSI. This will be accomplished via the newsletter and symposia sessions. I will attempt to get a condensed form of the information I receive into the newsletter as soon as possible after I receive it (see the following memo).

In addition to the ANSI participation, we would like to find a volunteer to serve as a liaison to the DEC internal standards working group. This will involve approximately six two-day trips to Maynard each year, participation at Symposia and submittal of articles to the newsletter. The goal for this position is to provide a bi-directional transfer of information and concerns between the SIG membership and DEC membership and DEC management and planning groups. If you are interested in this type of volunteer position, and if you can devote the time that will be required, contact me at the address:

Jim Flatten
304 Metallurgy
Ames Laboratory
Ames, Iowa 50011

There were three sessions presented at the St. Louis symposium that were related to standards issues. Don Feinberg of DEC led a discussion group about modeling systems. Todd Garcia from Tektronix talked about the proposed GKS standard, and Wade Scannell from Athena Systems Inc. and Don Feinberg teamed up to talk about viewing level software and GKS.

For the next symposia in Las Vegas, a session is planned for a panel discussion involving Jim Hodges, the DECUS representative to ANSI X3, and the DECUS

sponsored representatives to the various technical committees (MUMPS, BASIC, COBOL and GRAPHICS). This session will be a report to the membership of what is being done with the DECUS participation and why. Users will be given an opportunity to comment and to ask questions during the second half of the session.

Although the deadline for submissions will have passed by the time you read this, I want to solicit suggestions for possible future sessions. Some possibilities I have thought of are:

- VDI/VDM tutorial
- PMIG tutorial
- NAPLPS tutorial
- PHIG tutorial
- IGES tutorial
- PLP tutorial
- CORE vs GKS
- An introduction to the above alphabet soup

If you can speak on any of the above topics, or know someone who can, contact either myself or Richard Olds and we can get a session scheduled a future symposia.

The following is an excerpt from a memo sent to me by the DECUS X3 representative, Jim Hodges. I am including it because I think indicates why your participation is important.

Beginning approximately July 1, 1983, copies of pertinent ANSI-X3 documents will be forwarded to the appropriate SIGs for information purposes. The one person in each SIG will be the current ANSI participant (COBOL, BASIC, GRAPHICS, MUMPS). For other SIGs, it will be the newsletter, or if there is none the SIG chairman . . .

These documents will NOT be camera ready for your newsletter, rather they are raw data, for your information. Some will go to multiple SIGs. I personally do not plan to create copy for SIG newsletters, but I will submit some things for DECUScope.

I seek your advise on DECUS votes and DECUS positions in ANSI. Many things are marked "for future ballot" and these are things which you should develop positions on... I will mail my ballots ONE WEEK (7 days) before they close. Negative votes on ballots MUST contain a comment or reason...

Thanks
Jim Hodges

There are a couple of publications that are available now. The first is a packet of information about VDI/VDM standards efforts. The package includes:

- X3H3.3/82-45 Summary of Task Group Activities
- X3H3.3/80-34 The SD-3 for the Virtual Device Interface

X3H3.3/80-52 The SD-3 for the Virtual Device Metafile

X3H3.3/81-27 ANSI X3H3.3 Task Group Position Paper

X3H3.3/82-15 The dpANS for the Virtual Device Metafile

This package is available from CBEMA for \$15. The address for information is:

Ms. Debbie Shaner
X3 Secretariat, CBEMA
311 First St., NW, Suite 1200
Washington, DC 20001
(202) 737-8888

The second document is the GKS Draft International Standard document which is available from ANSI. There is a handling and reproduction fee of \$33 for this nearly 300 page document. Write or contact:

American National Standards Institute
Sales Department
1430 Broadway
New York, NY 10018
(212) 354-3300

Ask for ISO DP 7942 (GKS).

WANTED

Anyone with experience running ReGIS devices under RSX-11 please contact:

Aalan Rizzuto
ENC Controls, Inc.
P.O. Box 242
Cockeysville, MD 21030
(301) 667-4800

ACM/SIGGRAPH CONFERENCE

Bill Kramer

From July 25th to July 28th, I attended the annual ACM/SIGGRAPH conference in Detroit, Michigan. This is the premiere Computer Graphics meeting in the U.S., and was attended by over 14,000 people. It is impossible to relate in full detail all the activities of the week. Instead, I will try to provide a feeling for this type of meeting, since it is very different from the DECUS Symposium most of us are familiar with.

The conference is broken up into 2 parts, Courses and the Technical Program. 23 one and two day courses were presented on Monday and Tuesday of the conference week. They ranged from introductions to topics in computer graphics to technical "how-to" classes. Topics such as designing user interfaces, animation, the GKS standard, graphics and office automation, solid modelling, robotics and graphics for artistic designers were presented. I attended the two day course on "Developing Applications using GKS". Overall, the comments I heard from the attendees of other courses were positive and some were superlative. It is difficult to provide open classes without some assumption of the level of knowledge of the attendees. The main complaint I heard was that the classes tended to be too basic, because they tried to bring everyone up to the same level.

The GKS class was very good. It was a team effort by 4 members of the ANSI X3H3 committee, which is the Graphics Committee. The class was broken up into an overview of the current state of the hardware, an explanation of the standards process and an overview of GKS. The second day, we talked about some of the more advanced details of GKS, how to do input and user interaction with GKS and finally some thoughts on how to convert existing systems and packages (PLOT10, Core, Calcomp, etc.) to GKS.

The second part of the conference is called the Technical session. This is broken into several areas itself, and is much different than the DECUS format. There are only two presentations going on at anytime. One room is used to present groups of formal papers, and the other to hold panel presentations. All the formal papers are judged and selected before the conference by a committee, generally one out of every four submitted. These are then grouped by topic, and each author is allowed 20 to 30 minutes to present his findings. While the grouped topics are general, the papers themselves are very technical and specific.

The panel discussions are less formalized. From four to eight panelists present their views on current topics in computer graphics, such as standards, future directions in display technology, solid modeling and at the two "hot topics" this year: Japanese work in computer graphics and computer graphics education. In both types of presentations, the presentors are well prepared, and many use advanced visual presentations in film and video tape for demonstration. Much of the work discussed is aimed at high performance computer graphics - the type seen in the movies and on television, and not as common on the day to day life of most of us. When you attend SIGGRAPH, you have to remember that people who are there represent the leading edge of the Computer Graphics industry, and as such, there is a great deal of material which is not currently directly applicable to most computer users. But, just think if it were....

The next part of the Technical session is the Video Review, presented on Tuesday and Wednesday evening to a packed house. This is where the submissions which will eventually go out on the video tape copies of the review are seen for the first time. Some of the segments are spectacular. I found the most interesting segments to be the ones which described how an image was created. There was one which took a step by step look at how an image in the "Return of the Jedi" was created and filmed and another which explains recent work at the University of North Carolina. However, I am sorry to say those pieces were the exception, and most of the segments were simply one image after another. It was sometimes difficult for me to really appreciate the technical achievement of the segment because I knew so little of how it was made.

There was a fairly large art show this year, which consisted of over 100 pieces of computer generated art, which was static. These pictures were very diverse and interesting. Some were completely generated by computer, and others were created by using the computer as a tool. There was also a video film hall, with continuous showings of longer segments of video tape that could be shown at the video review. The film hall also had a number of computer enhanced pieces.

The last part of the technical program was the exhibit area. Over 185 individual exhibitors, from the smallest terminal vendor to DEC and IBM were present, showing all the products. I spent over 15 hours out the available 22 touring the floor and I am sure I still did not see half of the products. It is 10 times the DECUS exhibit hall in volume and diversity (but of course not in size since there are few mainframes and hands on exhibits.)

The exhibit hall is certainly the place to go if you are looking for any graphics products. It is useful to have an idea of what you are looking for before entering the exhibits, because it is very easy to be overwhelmed. There is only room to give a brief view of the new hot topics, and unfortunately, not being personally up to date on all areas of graphics, I can only mention areas I have followed (for example, I know very little about image processing and animation, and only a limited amount about CAD, so I can not attempt to review those areas.) There was a large interest in color graphics. Most new products were providing atleast 8 colors and a resolution of at least 512 pixels. Thankfully, people are turning to the problem of getting good quality color hardcopy as well, which resulted in an increase of ink jet printers and fairly low cost film processors. One product I have to mention is a plastic cone which fits over any 12 or 13 inch CRT and takes a picture of the screen. This is something many of us have made ourselves, but now the kit comes complete with a print camera and lens and a mounting bracket for a 35 mm camera. It is easy to use, and with the combination of instant color film, it provides instant hard copy. It is amazing it took so long to have a vendor "invent" this product.

Of course, everything is getting smaller, faster, more intelligent and cheaper. There are many terminals for under \$5000 which have 1024 X 1024 resolution and you can start getting some local intelligence after about \$7,000.00. Surprisingly, many graphics vendors are now making their terminals ANSI and/or VT100 compatible, which allows them to be used as regular timesharing terminal. Promises of even greater resolution are abundant with several Japanese monitor manufacturers showing CRTs with 1280 or 2048 Pixel resolution on a 19 inch screen. It was also interesting to note some new and old display technologies being introduced. In particular, several plasma displays were being demonstrated, even one from IBM. Also video digitizers seemed to be more common and of higher resolution, of course for less money.

Lest software be forgotten, the biggest development was several packages which conform to the ISO GKS standard, at various levels were announced. In particular, Tektronix is marketing one which conforms to the level 2B standard, which is basically complete output specification and requested input. It is a library written in FORTRAN 77.

Overall, the conference was extremely exciting. It provided a great opportunity to find people who are doing similar work in computer graphics, and also to get a little understanding into other areas of computer graphics. The courses were worthwhile and for those looking to make a purchase of graphics equipment, either software or hardware, it is the only place to be.

LAS VEGAS PRESYMPOSIA SEMINAR: EVERYTHING YOU WANTED TO KNOW ABOUT GKS IN ONE EASY LESSON

Summary: GKS, the Graphics Kernal System, is the proposed international computer graphics standard. Understanding GKS is important to everyone who is involved with Computer Graphics not only because it will be the approved standard in the near future, but because of a number of graphics products which will be based on GKS. This seminar will consist of 3 parts. First, an overview of computer graphics standards will bring everyone up to date on GKS, a number of standards which are currently being drafted and some future directions in standards. About 3/4 of the seminar will be covering GKS. This will include the benefits of GKS, how to use GKS and details on the proposed FORTRAN "binding" (specifics of subprogram and argument lists) of GKS. The final section will cover compatibility with ACM Core Graphics Standard, hints on converting from Core to GKS. This seminar should be attended by anyone who is writing or using modern computer graphics systems.

Speaker: Dr. Peter Bono serves as the chairman of American National Standards Institute Technical Committee on Computer Graphics and is the chief U.S. delegate to the International Standards Organization Working Group on Graphics. Dr. Bono was one of the eight authors of the original Core System and has been working with GKS since 1978. He has been involved with several implementations of the Core system. Thus, Dr. Bono is uniquely qualified to present an unbiased description of GKS, based on much experience with the practical, technical and theoretical aspects of standardization.

NEW ADDITIONS TO DECUS TAPE LIBRARY

new
VAX-45

NEWPOKER: Video Poker Game

Version: V1.0, July 1982

Author: Charles G. Davis, British Petroleum North America Trading, Houston, TX

Operating System: VAX/VMS

Source Language: BASIC

Memory Required: 248B

Special Hardware Required: VT100 or compatible terminal

NEWPOKER is a video poker game that play more like a real game than its predecessors. It uses a full deck of 52 cards, displays the cardsy of instead of telling you about them, and cannot see what is in your hand (until time to compare hands). It was created for VAX/VMS (any version), but with a few modifications will run on PDP's as well.

It works on any VT100 compatible terminal and uses little memory. Includes a documentation file to instruct the players, the source code (so you can tinker with it) and the executable.

Restrictions: Currently NEWPOKER can not "check" the bet.

Documentation on magnetic media.

Media (Service Charge Code): 600' Magtape (MA)

Format: VAX/ANSI (Blocked at 2048)

Keywords: Game
Category Index: 13
Operating System Index: VAX/VMS

new
VAX-47

revision
11-592

IRGL: Interactive Graphics Package

Version: V1.0, April 1983

Submitted By: Digital Equipment Corporation, Marlboro, MA

Operating System: VAX/VMS

Source Language: FORTRAN 77, FORTRAN VAX-11

Other Software Required: VAX-11 RGL

Special Hardware Required: VT125

IRGL is an interactive graphics package based on the subroutines that are part of Digital's VAX-11 RGL package. The primary intent is as a learning aid, however, functions have been added which make it useful in the preparation of graphics for presentations. It has been used successfully with DECUS No. VAX-42.

Documentation on magnetic media.

Media (Service Charge Code): Write-up (AA), 600' Magtape (MA)

Format: VAX/ANSI (Blocked at 2048)

Keywords: Graphics
Category Index: 8
Operating System Index: VAX/VMS

GIGIC2: GIGI Graphics Print-out on a Low-Priced C-I/OH Printer

Version: April 1983

Author: Jim-Son Chou, USC, Los Angeles, CA

Operating System: RT-11

Source Language: FORTRAN 11, MACRO-11

Memory Required: 4KW

Special Hardware Required: Two DLV-11J Serial Interfaces. The GIGIC hardcopy must be connected through a DLV11 RS232 interface to the computer with an address of 176510 (8). The C-I/OH is also connected through the RS232 interface to the computer. Its address is at 176500 (8).

GIGI2 is a program that allows the graphics on a DEC GIGI terminal to be dumped to a LSI-11 computer and then interpreted and printed on a low-priced C-I/OH printer. All one has to do is to run the GIGI2, then the entire screen content will be dumped to the printer. GIGI2 offers twice the size of the print out comparing to the previous version GIGIC.

Changes and Improvements: The size of the print-out is twice the one of the previous version, GIGIC.

Documentation on magnetic media.

Media (Service Charge Code): Floppy Diskette (KA) 600' Magtape (MA)

Format: RT-11

Keywords: Graphics
Category Index: 8

COBOL SIG

Table of Contents

	Page
Editor's File	58
Report from the COBOL SIG Standards Coordinator	59
Productivity Task Group Formed	63
RMS Wish List	64
Testing for Terminal Ready on an LA120	65
Building a Sharable COBOL Task under RSTS/E	66
Dynamic Loading of COBOL Tables	67

EDITOR'S FILE:

by Frank Corcoran, Reserved Word Editor

New Reserved Word Editor

At the Spring Decus Symposia in St. Louis I succeeded Don Lennert as editor of the COBOL SIG's newsletter, The Reserved Word. On behalf of the COBOL SIG, I extend our gratitude to Don for his work in behalf of the SIG and The Reserved Word.

MATERIAL FOR THE RESERVED WORD

I want to invite and encourage all users of COBOL on Digital machines to send material to me for publication in The Reserved Word. The goal of this newsletter is to provide a means of communication among COBOL users and a source of helpful hints and ideas for better use of COBOL on all Digital machines. The successful achievement of this goal depends on the participation of the COBOL SIG membership and the entire Digital COBOL User Community. Send in tales of your successes, your questions, your answers, your suggestions for improving The Reserved Word and anything else printable which relates to the use of COBOL to:

Frank Corcoran
M/S 23
Teradyne, Inc.
183 Essex St.
Boston, Ma. 02111

A SPECIAL APPEAL TO 36 BIT COBOL USERS

I want to make a special appeal to my fellow 10/20 COBOL users for material to include in The Reserved Word. In St. Louis, Digital announced its intention to abandon work on the long awaited Jupiter, a high end 36 bit machine and by inference the 36 bit architecture. In conjunction, DEC announced that future development on DEC-10s and DEC-20s will be directed toward achieving VAX compatibility. With this new background it is especially important that DEC-10 and DEC-20 COBOL users communicate with each other and voice their opinions concerning the future of 36 bit COBOL. The Reserved Word also can help us communicate our concerns to the COBOL SIG ;it, in turn, can help us communicate with DEC more effectively. Let's hear from you.

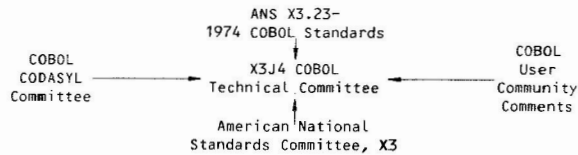
Editor's note: The following article was written by Dale Marriott in June, 1982 and, as a result, some of the date references may seem somewhat strange. However, I felt the contents of this article were of sufficient interest to all COBOL users to warrant its inclusion in this issue. The article following, titled "Postscript", was written by Dale in April 1983.

Report from the COBOL SIG Standards Coordinator

by Dale D. Marriott, COBOL SIG Standards Coordinator
Asst. Director, Systems & Programming
The Land Bank National Data Processing Center
7300 Woolworth Ave.
Omaha, Nebraska 68124

As the COBOL SIG Standards Coordinator, I interpret the duties of said position to be that of familiarizing myself with the needs and/or desires of the general membership of the COBOL SIG and then representing those members at the X3J4 COBOL Technical Committee meetings. From this simple position the plot thickens as one attempts to unravel the X3J4 Committee structure, its membership, policies and procedures, thereby ultimately establishing the next national standards for the COBOL Computer language.

The complex relationships, as I understand them, can best be illustrated by the following simplified graphic representation:



COBOL CODASYL Committee:

- Primary activity is the development of the COBOL computer language, which is recorded in the document, "Journal of Development" (JOD). The JOD is officially published every three years, with updates published several times per year.
- Composed of approximately 20 members, represented by half computer vendors and half computer users.
- This is a "working committee".

X3J4 COBOL Technical Committee:

- Primary activity is composed of two phases:
 - 1) The development of draft proposed American National Standard (dp ANS) X3.23 - 198x COBOL, which evolved through the X3J4 Technical Committee efforts employing the 1978 version of the JOD and the ANS X3.23 - 1974 COBOL Standard.
 - 2) The development of the recommended final version for the American National Standard (ANS) X3.23 - 198x, which evolves from the dp ANS

X3.23 - 198x COBOL and the comments received from the Public Review period of October 1981 through February 1982.

During this second phase, there has been the "fall - out" of two major categories of elements.

- a) Obsolete Language Elements, which represents those items substantially vendor-defined in current COBOL compilers and also supported in this new standard, which will be deleted from the standard in the next revision.
- b) Future Language Enhancements, are generally new concepts presented during the Public Revision period, which appear to merit further consideration, but based on existing information, do not demand inclusion in the new standard.

- The Committee is composed of approximately 24 members; half represent the computer vendors, the remainder representing computer users from industry, the Federal Government, and universities.

American National Standards Committee, X3:

- Primary activity is the formal adoption of a set of technical standards to be designated as the official COBOL language standards for the data processing industry (i.e ANS X3.23 - COBOL 198x).
- Pertaining to the establishment of the COBOL computer language standards.

COBOL User Community Comments:

- This activity officially took place after the "call for comments" by the X3 Committee, and is titled the ANSI Public Review period. After the dp ANS X3.23 - 198x COBOL version was presented to the X3 Committee, a majority vote forwarded the document for a four month public review, thus soliciting comments to improve this document. This official review period took place from October 16, 1981 through February 13, 1982. The results of this review period produced 2203 comments from the general data processing community, with approximately 50% being form letters, generated from a mailing campaign by two individuals.

My Observations:

During the five meetings of the X3J4 held between April 1982 and December 1982, the agenda will almost completely be taken up in addressing the 2203 public comments either as valid considerations for inclusion in this version of the standard, or in giving an explanation for their exclusion. It is my impression, from attending a single May 25 - 28 meeting, that the members of the X3J4 COBOL Technical Committee are sincere in their goal to recommend the "best" set of standards possible.

As your designated representative on this Committee, I would still

encourage you to obtain the dp ANS X3.23 - 198x from the X3 Secretariat at:

CBEMA
Suite 1200
1828 L Street, NW
Washington, D.C. 20063

With this document in hand, you can notify me of your concerns and I will then be better able to represent you during all decision votes on the continuing development of the final version of the ANS X3.23 - 198x COBOL Standard.

The official "Public Review Period" may be concluded, but as your Standards Coordinator, I continue to represent the DECUS COBOL SIG on the X3J4 COBOL Technical Committee. As the final version continues to evolve, your voices will be heard. Please mail your suggestions for improvement on the dp ANS document to the following address:

Dale Marriott, COBOL Standards Coordinator
Land Bank National Data Processing Center
7300 Woolworth Ave.
Omaha, Nebraska 68124

I wish to note that these comments represent my own impressions and do not necessarily reflect the official opinion of DECUS or the COBOL SIG Steering Committee.

Postscript by Dale Marriott, Standards Coordinator

The DECUS COBOL SIG has participated in the development of the second version of the DRAFT PROPOSED REVISED X3.J4 AMERICAN NATIONAL STANDARD PROGRAMMING LANGUAGE COBOL (dp ANS). We support acceptance of this second version as the new standard for the COBOL language.

We believe that the dp ANS comes closer than any prior language standard to achieving the purpose set forth by the X3.J4 Technical Committee:

To promote a high degree of machine independence in such programs in order to permit their use on a variety of automatic data processing systems.

Using the ANSI-74 COBOL standard as a foundation, the committee introduced several new features, which were defined in the CODASYL COBOL JOURNAL OF DEVELOPMENT 1978. The September 1981 version of the dp ANS was submitted for the first public review and comment period. Through this process, more than 2,000 public comments helped to enrich and improve the value of this proposed standard for the COBOL user community.

This revision of the dpANS has achieved the following milestones:

- Removal of numerous unnecessary restrictions
- Simplification of syntax rules

- Enhanced functionality for structure programming
- Expanded COBOL statement features that improve programmer productivity

The following examples illustrate a few of these expanded COBOL statement features:

a) INITIALIZE statement:

```
          01 GROUP-A.  
             05 AN-1      PIC X(3).  
             05 COUNT-1  PIC 9(4).  
  
STATEMENT  
INITIALIZE GROUP-A.  
INITIALIZE GROUP-A.  
REPLACING ALPHANUMERIC BY 'ABC'.  
INITIALIZE GROUP-A  
REPLACING NUMERIC DATA BY 1.  
RESULTS  
AN-1 = SPACES COUNT-1 = ZERO  
AN-1 = 'ABC' COUNT-1 is unaffected  
AN-1 is unaffected COUNT-1 = 1
```

b) Reference modification:

```
          01 AA  PIC X(6) VALUE "ABCDEF".  
          01 BCM PIC X(3).  
  
STATEMENT  
MOVE AA(1:3) TO BCM  
MOVE AA(5:) TO BCM  
RESULTS  
BCM = "ABC"  
BCM = "EF "
```

c) Structured Programming: Contained Program Feature

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. MAINPROGRAM.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
  
01 A-FIELD GLOBAL PIC X.  
  
PROCEDURE DIVISION.  
MAINPROGRAM-DRIVER.  
  
CALL "MOVE-DATA".  
STOP RUN.  
IDENTIFICATION DIVISION.  
PROGRAM-ID. MOVE-DATA.  
PROCEDURE DIVISION.  
  
MOVE 'X' TO A-FIELD.  
EXIT PROGRAM.
```

END-PROGRAM MOVE-DATA.
END-PROGRAM MAINPROGRAM.

d) Structured Programming: Scope Terminator feature

```
READ IN-FILE INTO W-REC AT END
  IF REC-CNT < 1 DISPLAY "ERROR"
  END-IF MOVE REC-CNT TO FILE-STAT
END-READ.
```

The COBOL SIG feels that the February 1983 version of the dpANS is worthy of becoming the new ANSI COBOL standard. We recommend that when the ANSI X3 Committee announces the public review period for this document, all SIG members review it, make comments to suggest improvements, and recommend that it be adopted.

The new dpANS should be available for purchase from:

X3 Secretariat:
Computer and Business Equipment Manufacturers Association
Suite 500
311 First ST., NW
Washington, DC. 20001

PRODUCTIVITY TASK GROUP FORMED

The COBOL SIG Steering Committee has formed a Productivity Task Group to promote the interchange of productivity ideas among DECUS members and between the COBOL SIG and the Commercial Language developers at DEC. This action was taken at the "Woods" meeting held March 25-26, 1983.

Keith Batzel was selected as the Chairman of this effort. Keith is a programmer/analyst for Crowe, Chezek and Company, a midwest CPA firm. He has five years experience in Data Processing, most of it using COBOL on PDP-11 and VAX computers.

MESSAGE FROM THE PRODUCTIVITY TASK GROUP CHAIRMAN

by Kieth Batzel

This project is of vital concern to all of us who work in or manage data processing shops where a commitment to using COBOL and related products exists

and will continue to exist for some time. COBOL is still the center of the business data processing world and it is this group's responsibility to insure that the increases in programmer and user productivity attributed to "4th generation languages" are acquired by COBOL and related products.

I am asking for volunteers to form a committee of persons committed to improving programmer and end-user productivity. We need a group with a cross-section of data processing experience such as manufacturing, distribution, accounting, research and development, government, education, etc. Please contact me if you are interested.

Participating in this project will provide a very special opportunity to work "hand in hand" with the product developers at DEC. Bruce Miller, who is development manager in the Commercial Languages Group, is the DEC counterpart to us. Bruce is responsible for COBOL and he is responsible for several programmer productivity products under development for which we can provide "user" input during design and development before "field" testing. Also, he is seeking our direction for future product offerings.

In addition, this group will need to promote changes in ANS COBOL. We can do this by working with the Standards Coordinator, Dale Marriott, on those changes which deal directly with productivity.

Finally, we must promote Symposia courses and "RESERVED WORD" articles about productivity methods and attitudes.

The objectives of the Productivity Task Group as well as its direction need to be defined by YOU. Please contact me as soon as possible at 219-232-3992 for further information. This committee needs to be functioning soon after the Spring Symposia.

RMS WISH LIST WANTED

by Chuck Evans
Director of Data Processing
Times Publishing Company
Times Square
Erie, Pa. 16534

In an effort to build better lines of communication with the developers of RMS, I am trying to start a RMS wish list. It would be under the auspices of the Data Management SIG because that seems to me to be the most logical one; however I would like to think of it as a cross-SIG endeavor.

Too many SIG wishes are responded to with the answer, "Well, that's really an RMS problem", and from there the wish disappears. I'm sure that this is as frustrating to the developers of RMS as it is to the users.

A session is planned for Las Vegas to discuss existing RMS wishes, and to accept new ideas and wishes. Please send any ideas or wishes you have to me at the address given above.

TESTING FOR TERMINAL READY ON AN LA 120

by Robert Crago
Senior Systems Programmer
Australian Broadcasting Commission
145 - 153 Elizabeth St.
Sydney, Australia
GPO Box 487
2001

We use several LA120-GAs on our DECSYSTEM-2060 for cheque printing applications and several years ago faced the problem of testing for terminal ready on our LA120s. However, we needed not only to be sure the LA120 was ready, but also that it had successfully printed each cheque. It is no use for the LA120 to merely XOFF the host when it had run out of stationary since the program had no way of knowing this had occurred. With numbered, accountable cheques, it was necessary for the program to record the number of each cheque in the master file for future enquiries. So on a change to a new batch of cheques, the program had to be informed of the starting number of the new batch of cheques and also perform alignment verification procedures before recommencing printing with the cheque that failed.

After some experimentation, we discovered that the LA120's answerback message was only transmitted (in response to an ENQ character) after the printer had successfully processed all characters sent to it previously. So adding an ENQ to the end of each cheque image caused the transmission of the contents of the answerback message (set to a single carriage return) after each cheque had successfully printed. The program waited after each cheque for input, timing out if nothing had been received in 10 seconds. If the timeout occurred, the cheque had not been printed successfully and the LA120's operator was asked to correct the problem. Apart from the code to await input or a timeout, this whole procedure was programmed in COBOL. LA120 parameter X is set to 0 to ensure the program never hangs in the XOFF'd state.

We recently attempted to convert to using LA120-RAs (i.e. without a keyboard) since we believed these were more suited to printing type applications. Incredibly, this model LA120 DOES NOT have the answerback facility (wasn't answerback invented for -RO printers to identify themselves back in the days of ASR-33s?). So we had to use the "product identification" sequence (with a longer sequence to add to each cheque) instead. That unfortunately was not the end of the problem. The LA120-RA also has no way of permanently taking it off-line (like the -GAs LINE/LOCAL button) short of holding down the offline button all the time. Thus on a stationary change, it starts printing again as soon as the lid is replaced. In fact, for this and other incompatibilities, we have been unable to get the same COBOL program to work successfully on a LA120-RA and have had to abandon the idea of using the "printer only" version of the LA120 in a printer only application!!

When will Digital realise that commercial customers have very different needs and start producing hardware that takes account of this? Monitoring DTR at the hardware level and optionally notifying the program of its absence would be a much clearer solution to this whole problem.

BUILDING A SHARABLE COBOL TASK UNDER RSTS/E

by Ron Washum
Yuma Regional Medical Center
2400 Avenue A
Yuma, Arizona 85364

This procedure requires the RSTS V7.0 MAKSIL feature patch 11.16.1 be installed allowing MAKSIL to handle multiuser tasks. This patch allows MAKSIL to strip the read-only code from the TSK, place it into a LIB file and link the two together. The LIB file may now be added as a resident library and shared by each concurrent TSK.

Our primary use of shared libraries at the Yuma Regional Medical Center is in the area of patient admissions where there is an average of six admitting tasks running concurrently. The total size of the task if this MAKSIL procedure is not used is 31K and with the average of six tasks running concurrently the total memory used is 186K. However, using this MAKSIL feature we have a sharable resident library of 7K and our six tasks of 24K for a total memory use of 151K; a savings of 35K.

The routine to accomplish this is as follows:

- 1) Compile your source.
- 2) Edit your task builder command file to include the /MU switch.

```
FILNAM.TSK/MU=FILNAM.OBJ/MP  
LIBR=RMSRES:RO
```

The /MU switch separates your task's read-only and read/write program sections. The read-only sections are placed in your task's upper virtual address space and the read/write sections in the lower virtual address space. This is the format required by MAKSIL to build a LIB file of read-only sections.

- 3) Run MAKSIL to split the TSK file into a LIB and TSK.

```
RUN $MAKSIL  
MAKSIL V7.0-07+/MU  
RESIDENT LIBRARY NAME ? FILNAM  
TASK-BUILD RESIDENT LIBRARY INPUT FILE <FILNAM.TSK>? <CR>  
INCLUDE SYMBOL TABLE (YES/NO) <YES> ? NO  
TASK IMAGE SIL OUTPUT FILE <FILNAM.SIL> ? FILNAM.TSK  
PROG BUILT IN 4K-WORDS, 0 SYMBOLS IN THE DIRECTORY
```

- 4) Add your resident library using UTILITY and your task is ready to run.

```
RUN $UTILITY  
UTILITY V7.0-07  
#ADD LIBRARY FILNAM.LIB/ADDR:??  
#EXIT
```

You may contact Ron Washum with any questions or problems concerning this procedure.

COBOL SUBROUTINE WHICH DYNAMICALLY LOADS A TABLE

by Dick Grilli
Teradyne Inc.
183 Essex St.
Boston, Ma. 02111

This article presents a subroutine which will dynamically load a table in the calling program from a specified file assigned to the logical name TBL:. This subroutine was developed on Teradyne's DECSYSTEM-2060 using COBOL-74. The complete source listing of this subroutine appears at the end of this article.

In order to call this subroutine the caller issues the following call:

```
CALL TBLD74 USING FILE-NAME, ERROR-CODE, CODE-LENGTH,  
ELEMENT-LENGTH, ELEMENT-TABLE, CODE-KEY,  
ELEMENTS-LOADED.
```

The subroutine arguments are defined as follows:

- 1) FILE-NAME is a PIC X(38) with usage is DISPLAY-7. FILE-NAME must have a period separating the file name from the file extension.

```
01 FILE-NAME PIC X(38) VALUE "STAFIL.SEQ" DISPLAY-7.
```

FILE-NAME is the name of the file you wish to use as an input file for the subroutine. It must be a sequential ASCII file, blocked zero. The record length must not exceed eighty characters. Each record must begin with the code-key that identifies the element to be loaded.

For example:

```
STC001VERMONT  
STC002NEVADA  
STC003MISSISSIPPI  
*****  
+ *  
CODE ELEMENT
```

- 2) ERROR-CODE is defined as PIC S9(10) usage is COMP.

```
01 ERROR-CODE PIC S9(10) COMP.
```

ERROR-CODE is a number returned by the subroutine if an error is encountered in attempting to load the table. The possible error code values are listed below:

- 1 = The specified input file was not found.
- 2 = Table overflow

3 = There were no matches throughout the file.

4 = There were one or more extraneous characters after the last character of the desired element.

5 = The record length of the input file was greater than eighty.

6 = The element length was less than zero.

7 = The code length was less than zero.

8 = The code length was greater than seventy nine.

Error codes of 1, 5, 6, 7 and 8 will cause the subroutine to return before any match attempts are initiated.

- 3) CODE-LENGTH is defined usage is INDEX. The CODE-LENGTH may be variable in length; however, it must not be less than zero or greater than seventy-nine.

```
01 STATE-CODE USAGE IS INDEX VALUE 3.
```

CODE-LENGTH is the length of the code that identifies the element to be loaded.

For example:

```
STC001VERMONT  
+++  
+  
CODE-LENGTH = 3
```

- 4) ELEMENT-LENGTH is defined usage is INDEX. The ELEMENT-LENGTH may be variable in length; however it must not be less than zero or greater than eighty.

```
01 STATE-LENGTH USAGE IS INDEX VALUE 17.
```

ELEMENT-LENGTH is the length of the element of the element to be loaded to the table. If the elements have various lengths, then the ELEMENT-LENGTH must be equal to the length of the largest element.

For example:

```
STC001VERMONT  
STC002NEVADA  
STC003MISSISSIPPI  
*****  
+ *  
ELEMENT-LENGTH = 14
```

- 5) ELEMENT-TABLE must be defined as DISPLAY-7. The aggregate length of the table can not exceed 5000 characters and optimally the length of the table should equal 5000.

```

01 STATE-TABLE      DISPLAY-7.
   05 STATES        OCCURS 250 TIMES.
   10 STATE-CODE    PIC X(3).
   10 STATE-NAME    PIC X(17).

```

6) CODE-KEY is defined PIC X(79) usage is DISPLAY-7.

```

01 MATCH-CODE      PIC X(79) VALUE "STC" DISPLAY-7.

```

CODE-KEY is a value for the code which identifies the element to be loaded. When a match occurs between the user's supplied CODE-KEY and the code at the beginning of each record within the input file, the element of that record will then be loaded ed to the table. All other records will be ignored.

For example:

```

      STC01VERMONT
      STC02NEVADA
      STC03MISSISSIPPI
      +++
      +
      CODE-KEY = STC

```

7) ELEMENTS-LOADED is defined PIC S9(10) usage is COMP.

```

01 STATES-LOADED   PIC S9(10) COMP.

```

ELEMENTS-LOADED is a number returned by the subroutine which is equal to the number of elements which have been loaded. This is useful for searching a table where the length is defined with an "OCCURS DEPENDING ON" clause.

Subroutine Source Code:

(Editors Note: This subroutine contains a call to a macro, LSTWRT, which was written by Teradyne's System Manager. This macro returns the date on which a file was last written to or an error message if the file was not found. It is used at Teradyne to check that a file exists before attempting an OPEN on that file. Readers who may wish to use this subroutine can replace this macro with an appropriate Declarative clause.)

```

/*****
IDENTIFICATION DIVISION.
*****
PROGRAM-ID.      TBLD74.

```

AUTHOR. DICK GRILLI.

INSTALLATION. TERADYNE DATA SERVICES.

DATE-WRITTEN. 17/11/81.

DATE-COMPILED.

```

*****          MODIFICATIONS          *****
* -- DATE --          --- CHANGE ---          -- PROGRAMER --
/
*****
* REMARKS:
* THIS PROGRAM IS A SUB-ROUTINE THAT WILL LOAD DATA FROM A FILE INTO
* A TABLE WITHIN THE CALLING PROGRAM.
* THE CALL AND ORDER FOR THE DATA-NAMES IS AS FOLLOWS:
* CALL TBLD74 USING FILE-NAME, ERROR-CODE, CODE-LENGTH, ELEMENT-LENGTH
* ELEMENT-TABLE, CODE-KEY, ELEMENTS-LOADED.
*
* THE FILE-NAME MUST HAVE A PERIOD SEPARATING THE FILE-NAME FROM THE
* FILE-TYPE. IT MUST BE A SEQUENTIAL , ASCII FILE, BLOCKED ZERO. THE
* RECORD LENGTH OF THE FILE MUST NOT EXCEED 80 CHARACTERS.
* EACH RECORD OF THE FILE MUST BEGIN WITH A CODE-KEY THAT IDENTIFIES
* THE ELEMENT TO BE LOADED. EXAMPLE: STC FOR A STATE-CODE. THE CODE
* AND ELEMENT LENGTHS MAY BE VARIABLE, HOWEVER, THE CODE-LENGTH MUST
* NOT BE LESS THAN ZERO OR GREATER THAN 79 AND THE ELEMENT-LENGTH MUST
* NOT BE LESS THAN ZERO.
* THE ERROR CODE PICTURE CLAUSE IS S9(10), COMP. THE ERROR CODE
* RETURNS MEAN THE FOLLOWING: 1=FILE NOT FOUND,
*                               2=TABLE OVERFLOW,
*                               3=THERE WERE NO MATCHES,
*                               4=ONE OR MORE EXTRANEIOUS CHARACTERS
*                               5=RECORD LENGTH > 80,
*                               6=ELEMENT LENGTH < ZERO
*                               7=CODE LENGTH < ZERO
*                               8=CODE LENGTH > 79
* THE ERROR-CODE RETURNS OF 1,5,6,7,8 WILL CAUSE THE SUBROUTINE TO
* EXIT, BEFORE ANY MATCH ATTEMPTS.
* THE ELEMENT-TABLE CAN CONTAIN UP TO 5000 CHARCTERS.
* THE AGGREGATE LENGTH OF THE TABLE IN THE CALLING PROGRAM CAN CONTAIN
* A MAXIMUM OF 5000 CHARACTERS.
* NOTE: IF THE TABLE IN THE CALLING PROGRAM IS LESS THAN 5000, OVER-
* WRITING MAY OCCUR, DEPENDING UPON THE AMOUNT AND SIZE OF THE
* ELEMENTS PASSED.
* THE CODE AND ELEMENT LENGTHS HAVE A PICTURE CLAUSE OF 'USAGE IS
* INDEX'. THE CODE-KEY MAY BE UP TO 79 CHARACTERS.
* THE ELEMENTS-LOADED IS A RETURNED NUMBER OF THE ELEMENTS LOADED TO
* THE TABLE.
* THE SUBROUTINE ARGUMENTS ARE:
*       01 FILE-NAME          PIC X(38)  DISPLAY-7.
*       01 ERROR-CODE        PIC S9(10)  COMP.
*       01 CODE-LENGTH       USAGE IS INDEX.
*       01 ELEMENT-LENGTH   USAGE IS INDEX.
*       01 ELEMENT-TABLE    DISPLAY-7.
*       05 TABLE-CHARACTER PIC X(1)    OCCURS 5000 TIMES.*
*       01 CODE-KEY         PIC X(1)    OCCURS 79 TIMS  DISPLAY-7.*
*       01 ELEMENTS-LOADED  PIC S9(10)  COMP.
*
* NOTE: IF YOU WISH TO INCLUDE COMMENT LINES WITHIN THE FILE BEING
* READ TO LOAD THE TABLE, BEGIN EACH LINE WITH AN ASTERISK (*).
* AS THE SUBROUTINE READS EACH RECORD IT WILL IGNORE THOSE

```

```

*          RECORDS BEGINNING WITH AN ASTERISK.          *
*****
/*****
ENVIRONMENT DIVISION.
*****
CONFIGURATION SECTION.

SOURCE-COMPUTER.          DECSYSTEM-20.
OBJECT-COMPUTER.         DECSYSTEM-20.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

        SELECT INPUT-FILE
              ASSIGN TO TBL
              ACCESS MODE IS SEQUENTIAL
              RECORDING MODE IS ASCII.

/*****
DATA DIVISION.
*****

FILE SECTION.

FD      INPUT-FILE
        VALUE OF ID IS FILID1
        RECORD CONTAINS 80 CHARACTERS
        BLOCK CONTAINS 0 RECORDS
        DATA RECORD IS INPUT-RECORD.

01      INPUT-RECORD          DISPLAY-7.
        05  RECORD-CHARACTER          PIC X(1)          OCCURS 80 TIMES.

/
WORKING-STORAGE SECTION.
*****

01  FILID1          DISPLAY-7.
    05  TAB-FIL-NAME          PIC X(6).
    05  TAB-FIL-TYPE          PIC X(3).

01  FIL-ID2          DISPLAY-7.
    05  FILLER          PIC X(4)          VALUE 'TBL:'.
    05  TAB-FIL-NAME          PIC X(6).
    05  FILLER          PIC X(1)          VALUE '.'.
    05  TAB-FIL-TYPE          PIC X(3).
    05  FILLER          PIC X(6).

01  FILE-DATE-AREA          USAGE IS DISPLAY-7.
    05  FILE-DATE          PIC X(17).

```

```

05  ERROR-FILE-DATE          REDEFINES FILE-DATE.
10  ERROR-RETURN          PIC X(1).
    88  FILE-NOT-FOUND          VALUE '*'.
10  FILLER          PIC X(16).
05  FILLER          PIC X(3).

*PROGRAM SUBSCRIPTS
*****

01  RECORD-INDEX          PIC S9(10)          USAGE IS COMP.
01  CODE-INDEX          PIC S9(10)          USAGE IS COMP.
01  TABLE-INDEX          PIC S9(10)          USAGE IS COMP.

*PROGRAM INDICATORS.
*****

01  ARE-THERE-MORE-RECORDS          PIC X(3)          VALUE 'YES'.
    88  THERE-ARE-MORE-RECORDS          VALUE 'YES'.
    88  THERE-ARE-NO-MORE-RECORDS          VALUE 'NO'.

01  IS-THE-TABLE-FULL          PIC X(3)          VALUE 'NO'.
    88  THE-TABLE-IS-FULL          VALUE 'YES'.

01  IS-THE-RECORD-A-COMMENT          PIC X(3)          VALUE 'NO'.
    88  RECORD-IS-A-COMMENT          VALUE 'YES'.

01  FIRST-CHAR-TO-LOAD          PIC S9(10)          USAGE IS COMP.
01  LAST-CHAR-TO-LOAD          PIC S9(10)          USAGE IS COMP.
01  FIRST-BLANK-CHAR          PIC S9(10)          USAGE IS COMP.
01  MATCH          PIC X(1)          VALUE 'Y'          DISPLAY-7.

*PROGRAM WORK AREA
*****

01  EXTRANEIOUS-CHAR          PIC X(1)          VALUE 'N'          DISPLAY-7.
01  RECORD-LENGTH          PIC 9(2)          DISPLAY-7.
01  MATCH-COUNTER          PIC S9(10)          VALUE 0          USAGE IS COMP.
/
LINKAGE SECTION.
*****

* SUBROUTINE ARGS.
*****

01  FILE-NAME          PIC X(38)          DISPLAY-7.
01  ERROR-CODE          PIC S9(10)          COMP.
01  CODE-LENGTH          USAGE IS INDEX.
01  ELEMENT-LENGTH          USAGE IS INDEX.
01  ELEMENT-TABLE          DISPLAY-7.
    05  TABLE-CHARACTER          PIC X(1)          OCCURS 5000 TIMES.
01  CODE-KEY          PIC X(1)          OCCURS 79 TIMES          DISPLAY-7.
01  ELEMENTS-LOADED          PIC S9(10)          COMP.

```

```

/*****
PROCEDURE DIVISION USING FILE-NAME ERROR-CODE CODE-LENGTH ELEMENT-LENGTH
ELEMENT-TABLE CODE-KEY ELEMENTS-LOADED.
*****/
A100-CHECK-FOR-INPUT-FILE.
    UNSTRING FILE-NAME DELIMITED BY '.'
        INTO TAB-FIL-NAME IN FILID1, TAB-FIL-TYPE IN FILID1.
    MOVE CORRESPONDING FILID1 TO FIL-ID2.
    ENTER MACRO LSTWRT USING FIL-ID2, FILE-DATE-AREA.
    IF FILE-NOT-FOUND
        MOVE 1 TO ERROR-CODE
        EXIT PROGRAM.

A110-CHECK-CODE-LENGTH.
    IF CODE-LENGTH < 0
        MOVE 7 TO ERROR-CODE
        EXIT PROGRAM.
    IF CODE-LENGTH > 79
        MOVE 8 TO ERROR-CODE
        EXIT PROGRAM.

A120-CHECK-ELEMENT-LENGTH.
    IF ELEMENT-LENGTH < 0
        MOVE 6 TO ERROR-CODE
        EXIT PROGRAM.

A130-CHECK-RECORD-LENGTH.
    COMPUTE RECORD-LENGTH = CODE-LENGTH + ELEMENT-LENGTH.
    IF RECORD-LENGTH > 80
        MOVE 5 TO ERROR-CODE
        EXIT PROGRAM.

A200-TABLE-LOADING-INIT.
    OPEN INPUT INPUT-FILE.
    MOVE 'YES' TO ARE-THERE-MORE-RECORDS.
    MOVE 'NO' TO IS-THE-TABLE-FULL.
    MOVE 'N' TO EXTRANEIOUS-CHAR.
    MOVE 0 TO MATCH-COUNTER.
    SET TABLE-INDEX TO 0.
    COMPUTE FIRST-CHAR-TO-LOAD = CODE-LENGTH + 1.
    COMPUTE LAST-CHAR-TO-LOAD = CODE-LENGTH + ELEMENT-LENGTH.
    COMPUTE FIRST-BLANK-CHAR = RECORD-LENGTH + 1.

A210-TABLE-LOAD-LOOP.
    PERFORM B010-MATCH-AND-LOAD-ROUTINE THRU B019-EXIT
        UNTIL THERE-ARE-NO-MORE-RECORDS OR THE-TABLE-IS-FULL.

A220-TABLE-LOAD-CLEANUP.
    IF MATCH-COUNTER = 0
        MOVE 3 TO ERROR-CODE
    ELSE
        MOVE MATCH-COUNTER TO ELEMENTS-LOADED.
    CLOSE INPUT-FILE.

A230-EXIT-PROGRAM.
    EXIT PROGRAM.

```

```

/
B010-MATCH-AND-LOAD-ROUTINE.
    READ INPUT-FILE
    AT END
        MOVE 'NO' TO ARE-THERE-MORE-RECORDS
        GO TO B019-EXIT.

    MOVE 'Y' TO MATCH.
    MOVE 'NO' TO IS-THE-RECORD-A-COMMENT.
    PERFORM C010-FORM-CODE-ROUTINE THRU C019-EXIT
        VARYING CODE-INDEX FROM 1 BY 1
        UNTIL (CODE-INDEX > CODE-LENGTH) OR (MATCH = 'N')
            OR (RECORD-IS-A-COMMENT).

    IF RECORD-IS-A-COMMENT
        GO TO B019-EXIT.

    IF MATCH = 'Y'
        ADD 1 TO MATCH-COUNTER
        PERFORM C020-LOAD-TABLE THRU C029-EXIT
            VARYING RECORD-INDEX FROM FIRST-CHAR-TO-LOAD BY 1
            UNTIL RECORD-INDEX > LAST-CHAR-TO-LOAD OR THE-TABLE-IS-FULL
    ELSE
        GO TO B019-EXIT.

    IF THE-TABLE-IS-FULL
        MOVE 2 TO ERROR-CODE
        GO TO B019-EXIT.

    PERFORM C030-CHECK-FOR-EXTRANEIOUS-CHAR THRU C039-EXIT
        VARYING RECORD-INDEX FROM FIRST-BLANK-CHAR BY 1
        UNTIL (RECORD-INDEX > 80) OR (EXTRANEIOUS-CHAR = 'Y').

B019-EXIT.
    EXIT.

/
C010-FORM-CODE-ROUTINE.
    IF RECORD-CHARACTER (CODE-INDEX) = '*'
        MOVE 'YES' TO IS-THE-RECORD-A-COMMENT
    ELSE
        IF RECORD-CHARACTER (CODE-INDEX) NOT = CODE-KEY (CODE-INDEX)
            MOVE 'N' TO MATCH.

C019-EXIT.
    EXIT.
-----
C020-LOAD-TABLE.
    SET TABLE-INDEX UP BY 1.
    IF TABLE-INDEX > 5000
        MOVE 'YES' TO IS-THE-TABLE-FULL
    ELSE
        MOVE RECORD-CHARACTER (RECORD-INDEX) TO TABLE-CHARACTER (TABLE-INDEX).

C029-EXIT.
    EXIT.

```

CO30-CHECK-FOR-EXTRANEIOUS-CHAR.
IF RECORD-CHARACTER (RECORD-INDEX) NOT = SPACES
MOVE 4 TO ERROR-CODE
MOVE 'Y' TO EXTRANEIOUS-CHAR.

CO39-EXIT.
EXIT.
Sample Source code for a program calling TBLD74:

IDENTIFICATION DIVISION.
PROGRAM-ID. SAMPLE.

.

DATA DIVISION.
WORKING-STORAGE SECTION.

01 FILE-NAME PIC X(38) VALUE "STAFIL.SEQ" DISPLAY-7.
01 ERROR-CODE PIC S9(10) COMP.
01 CODE-LENGTH USAGE IS INDEX VALUE 3.
01 ELEMENT-LENGTH USAGE IS INDEX VALUE 5.
01 STATE-CODE PIC X(79) VALUE "STC" DISPLAY-7.
01 STATE-TABLE DISPLAY-7.
05 STATE PIC X(5) OCCURS 1000 TIMES.
01 STATE-INDEX PIC S9(10) COMP.
01 STATES-LOADED PIC S9(10) COMP.

PROCEDURE DIVISION.

001-LOAD-TABLE.

CALL TBLD74 USING FILE-NAME, ERROR-CODE, CODE-LENGTH, ELEMENT-LENGTH
STATE-TABLE, STATE-CODE, STATES-LOADED.

100-FILE-CHECK.

IF ERROR-CODE = 1
DISPLAY "FILE NOT FOUND"
STOP RUN.
IF ERROR-CODE = 2
DISPLAY "THERE WAS A TABLE OVERFLOW"
STOP RUN.

IF ERROR-CODE = 3
DISPLAY "THERE WERE NO MATCHES"
STOP RUN.
IF ERROR-CODE = 4
DISPLAY "THERE WERE EXTRANEIOUS CHARACTERS IN THE FILE"
STOP RUN.
IF ERROR-CODE = 5
DISPLAY "ELEMENT LENGTH > 80"
STOP RUN.
IF ERROR-CODE = 6
DISPLAY "ELEMENT LENGTH < 0"
STOP RUN.
IF ERROR-CODE = 7
DISPLAY "CODE LENGTH < 0"
STOP RUN.
IF ERROR-CODE = 8
DISPLAY "CODE LENGTH > 79"
STOP RUN.

300-DISPLAY-ROUTINE.

PERFORM 400-DISPLAY THRU 409-DISPLAY-END
VARYING STATE-INDEX FROM 1 BY 1
UNTIL STATE-INDEX > STATES-LOADED.

DISPLAY STATES-LOADED, " : ELEMENTS LOADED IN TABLE".

STOP RUN.

400-DISPLAY.

DISPLAY STATE(STATE-INDEX), " HAS BEEN LOADED".

409-DISPLAY-END. EXIT.

OFFICE AUTOMATION SIG

OA SIG.....	page 78
From the Editor.....	page 79
Chairman's Chat.....	page 80
The Forum.....	page 81
Symposium Coordinates.....	page 90
Selling Office Automation to Management....	page 92
A Manager's View of Office Automation.....	page 95
Office Automation Cost Factors.....	page 100
New Frontiers in Training Technology.....	page 103
LA12/LA100 Functions and Features.....	page 108
12-Bit World.....	page 113

OA SIG STEERING COMMITTEE

Chairman:	Thomas J. Orlovski (703) 325-6161/2
Symposia:	
Coordinator	Pauline F. Kuntz (512) 691-7351
Presentation Recruiter	Fred Gerkins (201) 564-4095
Newsletter:	
Editor	Margaret Drake (512) 691-6105
Assistant Editor	Dwight H. Person (504) 527-5525
Assistant Editor	Ray Kaplan (602) 626-3116
12-Bit Editor	Bob Hassinger (617) 435-3452
Area Coordinators:	
Personal Computers	David E. Soloman (313) 769-8500
Standalones	Loretta Ammons (916) 752-3524
Professional/Knowledge Worker	Tom J. Green (415) 422-6231
SIG Liaison:	
Graphics	Bill Kramer (302) 453-6031
VAX	Douglas J. Wilson (617) 253-2041
DIBOL and 12-Bit	Lawrence H. Eisenberg (213) 788-0354 Ruth K. Koolish (312) 724-9280 Bob Hassinger (617) 435-3452
DEC Counterpart	Dick Hill (603) 884-3960

FROM THE EDITOR

Welcome to the third issue of the DECUS OA SIG newsletter. This is the first issue that appears in combination with newsletters for Graphics, DIBOL, and COBOL SIGS and the first issue under the new paid subscription plan.

There are other firsts---this is the first issue of the newsletter not under the able direction of Clyde Kelley. As Clyde stated in the last issue, he has moved from MIT to the Micro Electronics Center of North Carolina in Research Triangle Park. Our best wishes go with him and our thanks for his work.

Please remember that this is your newsletter and can only be of use to the Digital OA Community if we have your input and suggestions for topics, articles, and features.

Please send all material to:

Margaret Drake
Director of Institutional Management Support
The University of Texas Health Science Center
7703 Floyd Curl Drive
San Antonio, Texas 78284

(512) 691-6105

Please call, write or see me in October in Las Vegas. Thanks!

Chairmans's Chat

---Tom Orłowski

With St. Louis behind us, the OA SIG is firmly established and growing rapidly. This growth reflects the continuing expansion of interest in office automation and the thirst for knowledge related to that very nebulous arena.

The SIG is responding to increased membership and interest by providing this newsletter forum, establishment of working groups like the recently formed All-In-One group, and presenting a wide variety of OA seminars at the Las Vegas Symposia. As you can see, the SIG is trying to use a variety of means to keep you informed and provide contacts that might be able to help you. This is your special interest group; let us try to make it available for you.

I know you all join me in saying thanks and farewell to our first DEC counterpart, Eileen Deal. She has been with us from day one and what we have today is, in large measure, due to her hard work. Unfortunately, her corporate duties have increased to the point where she no longer has time to devote to the SIG. Thanks a million for all of your efforts. Good luck in your future endeavors.

See you in Las Vegas.

Tom

THE FORUM

(Editor's Note: The following material ended up in Tom Orłowski's hands; he does not know the originator--will you please let us know who you are? The material is super.)

EVALUATION OF "USER FRIENDLY" FACTORS IN AUTOMATED OFFICE SYSTEM SOFTWARE

Each of the 8 factors listed below has been shown to have a strong impact on the productivity that can be obtained in the automated office.

FACTORS	DEFINITION
1. CONSISTENCY	Common behavior and commands throughout all the user functions. BENEFITS: Reduces learning time. Reduces mistakes.
2. SAFEGUARDS	Protect the user from costly accidents and mistakes. BENEFITS: Reduces time lost correcting errors. Increases confidence in system, encouraging use.
3. RESPONSIVENESS	Common operations are rapidly executed. Response times for an operation are consistent. BENEFITS: User can work at his fastest pace. User can plan his work time.
4. INTEGRATION	User actions are not restricted to a particular function or application. BENEFITS: Speeds execution of work by allowing immediate access to all commands required.
5. RESUMABILITY	User can stop one task, perform another task under a different function and then return to the place he left in first task. BENEFITS: Allows user to carry on several tasks at on time-

moving easily among them as the work requires. User's responsiveness to a new situation is not penalized by forcing him to lose time or data.

- 6. SIMPLICITY Natural, uncomplicated projection of the system. "Transparent".
BENEFITS: Easier to learn. User can apply his knowledge of task without learning details of system.
- 7. FLEXIBILITY System allows the user to customize options.
BENEFITS: Allows all types of users to maximize productivity with the system.
User can control his work.
- 8. FEEDBACK User is kept clearly informed about status of his request, and the options available to him.
BENEFITS: User does not waste time checking manuals or calling for system status.

FACTORS ARE DIVIDED INTO THREE CATEGORIES

The following pages list these factors in more detail and illustrate how features would be classified in one of the following three categories.

- UNACCEPTABLE - No expected productivity gain from office automation system (buyers should avoid systems with factors that fall into this range).
- ACCEPTABLE - Expect some productivity gain (Office automation systems with factors at this level are becoming widely available).
- SUPERIOR - Expect great productivity gain compared to manual system (Office automation systems with factors at this level are unusual but will become more common in the future).

SUGGESTED RATING SYSTEM FOR "USER FRIENDLINESS"

The rating method below can be applied to a factor, either in a system of automated programs or in an individual application program.

FIRST - DETERMINE SPECIFIC SOFTWARE (APPLICATIONS) TO BE RATED

Divide system into its application programs

Example: Word Processing, Electronic Filing, Record Processing, Calendaring, Electronic Mail, Decision Support, Voice, Graphics

SECOND - RATE EACH SOFTWARE APPLICATION USING THE 8 "USER FRIENDLY" factors

1. Assign scores to the components of each factor as related to each application

unacceptable -1
 acceptable +1
 exceptional +2

2. Assign a weight to the subcomponents of each factor which are most critical for your intended use.

3. Multiply the rating of the component factor by its weight.

4. Add totals for each factor.

5. Total all factors for each application.

THIRD - RATE THE SYSTEM (If applicable)

1. Weight applications that are most important to your intended use.

2. Multiply the applications total factor score by the weighted value of the application.

USER FRIENDLINESS IN AUTOMATED OFFICE SYSTEMS SOFTWARE

1. CONSISTENCY -

a. The same commands are used for the same action within each function and across all applications.

Example: A backspace key used to delete within text editing performs the same function in electronic mail

Example: P used for PAUSE in one application and for PURGE in another

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Seldom	Frequently used commands have the same name	Always

No commands are used destructively in one application and constructively in another.

2. SAFEGUARDS -

a. Error Correction is fast and simple.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No, must rekey entire entry line again	Yes, can correct before sending by rekeying only the error	Yes, can stop and enter a correction even after processing has begun

b. Data cannot be accidentally destroyed.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Data can be destroyed with no warning	User is warned before data is destroyed and given a chance to save it.	Automatic copy kept of all information user enters or generates

c. User can reverse the results of last action and return to the original state of the document or file.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Cannot reverse	Can reverse by rerunning	Can "undo" last action

3. RESPONSIVENESS -

a. Are common operations rapidly executed?

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No. Average exceeds 2 seconds a command	Yes. Average under 2 sec a command	Yes. Average under 1 sec a command

b. Does the same operation have a consistent response time?

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No	Usually	Always

c. Is user warned if response time will be longer than average?

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No feedback	No, but system gives periodic signals that it is working on users request	Yes, and informs user when response time will exceed the average. Also gives reports as to what step is in process.

4. INTEGRATION -

a. Commands are integrated within an application.

Example: Within the word processing application the user can insert when creating, editing or printing a document.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Seldom	Usually	All reasonable commands are integrated

b. Functions are integrated across applications.

Example: User can use editing function from word processing when he is in electronic mail.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Seldom	Usually	All reasonable commands are integrated

c. Documents and files can be transferred from one application to another.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No, files created by one application cannot be used by another	Yes, but must save file and then reload it into the next application	Yes, all reasonable transfers can be done directly

d. Data entered under one application can be passed to a file created under another application.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No	Yes	Yes, and can transfer automatically once a link is established (Ex. Data entered into a spreadsheet immediately updates a graph)

5. RESUMABILITY -

a. The user can stop one task and perform another task under a different function.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No, a task cannot be interrupted	Yes	Yes, and user can resume his task from the place he left Many actions can be at different stages--user can move freely among them

6. SIMPLICITY -

a. Task steps are in a natural sequence.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Seldom	Usually	Always

b. Number of steps the user must specify to perform a task.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
User specifies steps to do task in detail	Many steps are automatically performed by the system.	All steps are automatically performed by the system.

c. Defaults are provided for input/output parameters.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Seldom	Usually	Always

d. User can redisplay choices prior to executing a task.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No	Yes, but only just prior to execution	Yes, at any point prior to execution.

e. Recognizable terms and images are used in instructions.

Example: outline of page is displayed rather than a blank screen

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Seldom	Usually	Always (includes use of graphics and instruction by example)

7. FLEXIBLE -

a. User can define own macro commands.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No	Sometimes	Always (User can program an entire task sequence if desired)

b. User preferences can be stored and activated.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No	Yes, but user must activate	Yes, and system automatically activates when user signs on

c. Menu can be bypassed by user.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No	Yes	Yes, and several commands may be entered at one time ("stacked")

d. Provides more than one level of prompts, help and error messages

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Seldom	Usually	Always

e. User can override system parameters.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Seldom	Usually	Always

8. FEEDBACK -

a. Error messages describe problem in English.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Error messages use code numbers	Yes	Yes, and system suggests appropriate user actions

b. System provides feedback to user actions.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
Seldom	Usually	Always

c. Responses are polite and friendly.

UNACCEPTABLE	ACCEPTABLE	SUPERIOR
No, responses are abrupt and/or judgmental	Usually	Always

d. Explanations use computer terminology.

UNACCEPTABLE Seldom	ACCEPTABLE Usually	SUPERIOR Always
------------------------	-----------------------	--------------------

e. Help is available for each command and function.

UNACCEPTABLE Not available- or can only be accessed from main menu	ACCEPTABLE Yes, but user must identify help message needed	SUPERIOR Yes, and system displays appropriate help message automatically
--	--	--

f. System prompts user for required parameters.

UNACCEPTABLE No	ACCEPTABLE Yes	SUPERIOR Yes, and system provides name and format of required parameters
--------------------	-------------------	--

HELP ME PLEASE

William Barlow is looking for a copy of the program "RTFLOP" from OS/8. Anyone who can provide a copy on RX01 media should contact him by writing to: PO Box 33, Portland, OR 97201 or by telephone: (503) 224-1030.

He is also looking for a copy of the operating system with documentation for "PASCAL-S" on RX01 media for a PDP-8/A in the DEC Data System 310.

Symposium Coordinates

---Pauline Kuntz

From the enthusiastic wrap-up session in St. Louis, I returned to San Antonio knowing we would have a "Texas sized" group of submissions for the Las Vegas symposium and I was right.

The "goodness of fit" exercise took place in Marlboro during July. When 19 people, working together for the first time, negotiated, cancelled, traded, combined, and scheduled the symposium into the time and space available.

Thirty-seven of the 65 OA sessions were scheduled. OA and DIBOL will share a 200 person room for most of the sessions. However, some of the OA sessions are scheduled in an even larger room. For example, The All-In-One Performance Update and the All-In-One Question and Answer sessions are scheduled Thursday evening in a 400 person room. All of the OA session rooms are adjoining on the 3rd floor - west tower of the MGM Grand.

The pre-symposium on Sunday, October 23, 1983, features two OA sessions, Design of Applications for the "Professionals" and The Human Factors in System Design (Ergonomics). Some of the titles of the Monday through Friday, October 24 - 28, 1983, sessions are:

- Videotex What? Office Automation
- Standardized Software Generator for Office Automation
- OA Ergonomics
- Timesharing in OA
- OA Cost Factors
- All-In-One/USAA
- DECmate Word Processing for Data Processing
- Current Legal Implications
- Introduction OS/78 and COS 310 on the DECmate II
- MASS 11
- EasyLink: Communicate with the world
- All-In-One Overview and Applications Portfolio
- Message Router
- Business Graphics
- DECType Wordprocessing on the VAX

Monday evening 8:00 PM to 10:00 pm, an OA Magic Session is scheduled. Phillip Swanson, All-In-One Working Group Chairman, invites all interested people to participate. Please come to DECUS prepared to share the tricks and quirks that you have discovered in your endeavors to automate the office. Presentations can include:

Categories	Subjects
Would you believe?	DECmate
Unsolvable Problems	Electronic Mail
Solutions to Unsolvable Problems	All-In-One
	Graphics
	Professional 325-350
	Word Processing
	Desk Management

A small prize will be awarded for the best tall tale in each of the categories. An overall grand prize, determined by an impartial panel of judges, will be awarded.

Another meeting of special interest will be the DECWORD and Word 11 sharing of information. Please contact Eric Moothart for additional information.

A number of related to OA sessions (39) are identified in the cross reference to assist you in planning your session attendance schedule for the week.

Thanks to all the people who submitted abstracts and a plea to others to do so for the Cincinnati symposium, May 1984. Also, I solicit your help in identifying sessions you want to hear in Cincinnati. Please respond BEFORE February 1, 1984. You can reach me by telephone (512) 691-7351 or at the following address: The University of Texas Health Science Center, 7703 Floyd Curl Drive, San Antonio, Texas 78284.

SELLING OFFICE AUTOMATION TO MANAGEMENT OA: A PEOPLE OR A TECHNOLOGY ISSUE

by
James McClure
Comp-U-Staff
1 Investment Place
Towson, Maryland 21204
(301) 828-0788

(Editor's Note: These are visuals from Jim's presentation in St. Louis.)

I. Why All the Fuss?

a. Demand	Supply
Increased office workload	Smaller/quicker logic
Flattening of labor supply	Bigger/faster storage
Rising labor costs	Broader/faster communications
Employee expectations	Improved software/firmware techniques
c. The need to improve office labor performance	The ability to deliver computer support at a reasonable cost
d. A convergence of demand and supply	

II. Interesting Trends in the Office

a. Overhead costs
1960's 20-30%
1970's 40-50%
b. Cost of business letter
1965 \$2.44
1975 \$3.79
c. Equipment investment per employee
1975 \$2000/year
1985 \$8000/year

(Blue collar investment in 1977-\$25,000/year)

III. Employment by Major Occupational Group

Workers	(Millions)		DOT	
White-collar	44.8	50%	68,500	92%
Blue-collar	29.5	33%	6,000	8%
Service	12.3	14%	N.A	N.A.
Farm	2.7	3%	N.A	N.A
TOTAL	89.3	100%	74,500	100%

Office workers account for 66% of national payroll cost

IV. The Automated Office

Definition: The systematic approach to using computer support to perform traditional office function, i.e. "MULTIFUNCTIONAL" office computing

V. Identifying Management Needs

- a. Increase productivity in work units
- b. Optimize utilization of all resources
- c. Implement positive change with least amount of disruption
- d. Improve product quality while maintaining costs
- e. Protect political status and turf
- f. Maintain proper management controls

VI. Overcoming Management Objections

- a. Analysis of human factors and change implications
- b. Providing an approach to managing technological change
- c. Providing a smooth implementation approach
- d. Information power - giving management access to key information

VII. Obstacles to Implementation

- a. Fear of change
- b. No one vendor system offers comprehensive set of capabilities
- c. Frenzy in market has users confused
- d. Improper planning and implementation
- e. Inadequate post implementation evaluation

VIII. Why Now?

- a. Climate is right
- b. Technology is available
- c. Payoff can be significant

IX. Where Will All This Lead Us?

- a. Access to computer power at every desk
- b. Every office is a link in a telecommunication network
- c. Organizational data base is universally available
- d. Faster access to information and co-workers
- e. More paper throughput
- f. More administrative power to more people

MORE COST-EFFECTIVE BUSINESS

X. The Key to Success

OA must be regarded primarily as a PEOPLE issue with technological considerations.

A MANAGER'S VIEW OF OFFICE AUTOMATION OA: A PANACEA OR A MANAGEMENT TOOL

by
James R. McClure
Comp-U-Staff
1 Investment Place
Towson, Maryland 21204
(301) 828-0788

(Editor's Note: These are visuals from Jim's presentation in St. Louis.)

I. Interesting Trends in the Office

- a. Overhead costs
1960's 20-30%
1970's 40-50%
- b. Cost of business letter
1965 \$2.44
1975 \$3.79
- c. Equipment investment per employee
1975 \$2000/year
1985 \$8000/year

(Blue collar investment in 1977-\$25,000/year)

II. The Automated Office

Definition: The systematic approach to using computer support to perform traditional office function, i.e. "MULTIFUNCTIONAL" office computing

III. Identifying Management Needs

- a. Increase productivity in work units
- b. Optimize utilization of all resources
- c. Implement positive change with least amount of disruption
- d. Improve product quality while maintaining costs

- e. Protect political status and turf
- f. Maintain proper management controls

IV. MAJOR MANAGEMENT CONSIDERATIONS/CONCERNS

- a. Productivity
- b. Cost effectiveness
- c. Ease of installation and use
- d. Impact on employee jobs
- e. Impact on organization structures
- f. Impact on manager's job tasks
- g. Impact on manager's status
- h. Management control methods

V. MAJOR ISSUES

- a. Change
 - Manager
 - Employee
- b. Technology
 - Hardware
 - Software
 - Communications
- c. Planning, integration and assimilation into work environment

VI. Overcoming Management Objections

- a. Analysis of human factors and change implications
- b. Providing an approach to managing technological change
- c. Providing a smooth implementation approach
- d. Information power - giving management access to key information

VII. Obstacles to Implementation

- a. Fear of change
- b. No one vendor system offers comprehensive set of capabilities
- c. Frenzy in market has users confused
- d. Improper planning and implementation
- e. Inadequate post implementation evaluation

THE MOST IMPORTANT CONTRIBUTOR
TO A SUCCESSFUL PROCUREMENT:

USER INVOLVEMENT

THE BEST MEASURE OF
THE SUCCESS OF A PROCUREMENT:

USER ACCEPTANCE

VIII. MAJOR STEPS IN PROCUREMENT

- 1. Prepare SPECIFICATIONS
- 2. Select EVALUATORS
- 3. Develop EVALUATION PROCEDURE
- 4. Conduct EVALUATIONS
- 5. Coordinate INSTALLATION

- 1. PREPARE SPECIFICATIONS
 - List user requirements
 - Learn equipment characteristics
 - Determine mandatory requirements
 - Determine desired features
- 2. SELECT EVALUATORS
 - Evaluation representatives
 - Operating characteristics evaluators
- 3. DEVELOP EVALUATION PROCEDURE
 - Mandatory requirements
 - Desired features 35%
 - Price 35%
 - Operating characteristics 30%
- 4. CONDUCT EVALUATION
- 5. COORDINATE INSTALLATION
 - Supplies
 - Dedicated power
 - Training
 - User groups

IX. Where Will All This Lead Us?

- a. Access to computer power at every desk
- b. Every office is a link in a telecommunication network
- c. Organizational data base is universally available
- d. Faster access to information and co-workers
- e. More paper throughput
- f. More administrative power to more people

MORE COST-EFFECTIVE BUSINESS

X. Why Now?

- a. Climate is right
- b. Technology is available
- c. Payoff can be significant

PACING FACTORS - TECHNOLOGY

- Vendor/demand-driven
- Development lead time
- Integration (broad function)
- Pricing (proliferation)
- Friendliness
- Generalized software
- Technological obsolescence
- Standardization

PACING FACTORS - HUMAN CONSIDERATIONS

- Acceptance and assimilation of change
- Resistance/intimidation
- Organization
- Fragmented responsibility
- Competition/duplication/lack of leadership
- Innovation
- User involvement
- Broadened perspective
- Technical support
- Changing perspective
- Emphasis on support

PACING FACTORS - MATURATION PROCESS

- Experimentation
- Tinker with technology - raise awareness
- Pilot
- Test assumptions - master the tools
- Selective implementation
- High payoff applications - innovations
- General propagation
- Taken for granted - can't afford to be without e.g. copiers, computers, calculators...etc.

XI. THE BOTTOM LINE

OA must achieve ORGANIZATIONAL, EMPLOYEE, and INDIVIDUAL MANAGER goals without major disruption to the organization.

XII. THE KEY TO SUCCESS

OA must be regarded primarily as a PEOPLE issue with technological considerations.

Office Automation Cost Factors

Chip Wyser
Digital Equipment Corporation
Merrimack, NH

Reported by Phil Beene, DECUS Scribe Service

(Editor's Note: This material resulted from the "Scribe" service used at the St. Louis symposium. Look for more of this helpful service at Las Vegas.)

Measuring the cost and efficiency of an office can sometimes be a difficult task. A task which some office managers find even more difficult, however, is justifying to top management the cost of converting to office automation.

During the recent DECUS Symposium held in St. Louis, office automation users received a presentation on how to justify the cost of implementing office automation by Chip Wyser, an office automation manager representing DIGITAL.

Wyser opened his discussion by stressing the magnitude of office costs in the business world. Quoting from the "Harvard Business Review," Wyser said American businesses spent over \$one trillion on office supplies in 1982. He says this figure is expected to increase to almost two trillion by the 1990s.

With over 50% of America's workforce employed in an office environment costing businesses trillions of dollars, Wyser says more business people will have to switch to efficient automatic office systems if they want to remain productive in our competitive society.

According to Wyser, there are five goals which must be determined before implementing automation in any business:

1. Determining the company's critical success factors.
2. Understanding how the company is organized.
3. Determining the size of the office group involved.
4. Understanding how to communicate within the office group.
5. Knowing what results are desired.

Wyser said that if an office manager properly determines these goals and implements his program correctly, there are four major gains which can be expected from office automation:

1. EFFICIENCY. Receiving the same output at a lower cost.

2. EFFECTIVENESS/PRODUCTIVITY. Receiving increased output at the same cost.
3. ORGANIZATIONAL DESIGN. Gaining increased output at a lower cost.
4. QUALITATIVE BENEFITS. Generally known as keeping a competitive edge.

During the next portion of his presentation, Wyser walked through the latter four user benefits, and explained each in detail.

According to Wyser, the various degrees of efficiency and productivity are determined through successful office organization. He said office automation can make an office more efficient by saving employees' working time. This goal can be achieved by automatic features such as text processing, reduced ancillary costs and waterfall analysis.

Waterfall analysis, Wyser says, is a term used to describe the breakdown of actual employee work time. By graphing an individual employee's workday, Wyser was able to demonstrate how much time is lost during the average office worker's day. According to surveys, the average engineer spends only 33% of his day doing the actual research he or she is paid to do. The rest of the time is spent writing letters, reading memos, talking on the phone and traveling. With the average business letter costing around \$7.60, office automation features such as electronic filing, mailing and copying will help save company money and the time of everyone from clerical help to professional managers. These results alone are powerful proof system managers can use to persuade management to invest in office systems, Wyser explained.

Although they are more difficult to quantify, qualitative gains, Wyser said, are often the greatest benefits offered to an office through an efficient automated office system.

After the automated system has saved business management valuable time by eliminating paperwork, travel and phone calls; managers are able to spend more time evaluating problems and making well-thought decisions, Wyser said. Not only will this result in better organization and increased efficiency, Wyser says, it will create an increased knowledge worker capability and allow workers more time to run their business properly.

To achieve these results, Wyser says, a system manager must create a strategy which will convince his or her boss to invest in a better integrated automatic office system. Listed below is Wyser's suggestion for outlining your own strategy:

1. The most essential step to success is obtaining the support of top management. Without it, you will not succeed in creating interest amongst the people who will determine whether funds are made available for investment.
2. Management should be presented with solid data which will be accurate and convincing. This could include: a) nominal/department functions, b) sample/global survey, c) interviews.

3. Establish break-even. According to Wyser, the cost of a system is low when a business adds the various benefits, write-offs and depreciation available. The final cost can often be proportionately low.

Bringing his discussion to a close, Wyser gave a summary of the advantages of an automatic office system. Stressing good strategy, strong data and the support of top management, Wyser gave a convincing presentation to users that they too can soon have their own management investing in new automated office systems.

NEW FRONTIERS IN TRAINING TECHNOLOGY

Del Lippert
Digital Equipment Corporation
Bedford, MA

Reported by J. Rick Mihalevich, DECUS Scribe Service

(Editor's Note: This material resulted from the "Scribe" service used at the St. Louis symposium. Look for more of this helpful service at Las Vegas.)

Del Lippert of DIGITAL stated that the next ten years promise to be a real challenge both in education and in industry. Rapid technological change and the increasing intensity of foreign competition will test our businesses. If we react boldly, we can provide the kind of training and retraining environments that will keep our organizations and our employees competitive.

He went on to describe some of the factors which are leading us to choose educational technology as part of our training solution in industry. He also described how DIGITAL applies educational technology in an industrial training environment, since their training needs are similar to those of many industrial organizations. Finally, he discussed some exciting new developments in educational technology.

Why do we need to apply technological solutions to our training needs in industry? The answer to this begins with one word: change. As a recent report from the Office of Technology Assessment of the U.S. Congress states,

"A key element in all of these educational needs is that they will constantly change. In a rapidly advancing technological society, it is unlikely that the skills and information base needed for initial employment will be those needed for the same job a few years later."

The people hired today will not be performing the same job in five years, or even two years in many cases. The problem is that the working lifetime of our employees is now very much greater than the development and support lifetimes of new technologies.

This means that most employees will need to be retrained more than once during their careers in order to remain productive. For example, journeyman electricians in the automobile industry must now be retrained in electronics, robotics, and computers. The old model of work in a trade or profession is a long apprenticeship leading to an indefinite and stable journeyman status. This model based on the medieval guilds is still accepted by many people today. Change is breeding change and we can no longer expect that the obsolescence of skills facing our country today will be a unique event.

Right now a typical employee can look forward to a career that consists of cycles with alternating periods of training and productive contribution. It used to be that the productive period took up most of a person's working life. Childhood schooling and sometimes college were sufficient preparation for a lifetime of work; this is no longer the case.

The typical engineering graduate in this country is 22-23 years old. Engineers themselves state that they face their greatest challenge around age 27, and they categorize themselves as technologically obsolete by age 32. That is an interesting number, because in 1982 the median age of the United States' population was 32. Compare this with the situation projected for 1990, when the median age will be 40.

The challenge facing us is to keep those who will soon amount to most of our workforce productive in a time of great technological change. This is more than a problem of higher management. While good technical people can overcome poor management, the reverse is not generally true.

Compounding this problem is the fact that many of the qualified applicants to engineering schools were rejected in 1980 due to lack of capacity in these schools. Because it will become increasingly difficult during the next decade to find qualified engineering graduates, ongoing retraining must be maintained as a source of productive technical work force.

The concept of a learning cycle that keeps people both challenged and productive is one answer to these problems. The learning cycle begins with formal schooling, perhaps college, and sometimes an advanced degree, all provided by traditional academic institutions. The cycle is maintained by continued retraining to meet both job needs and the need for personal growth. This continually changing part of the learning cycle is increasingly the task of industry itself or of industry in partnership with educational institutions.

The reliance on the learning cycle is based partly on the faith in the adaptability of adults and partly on recent psychological studies. Adults, like children, continue their intellectual development beyond the age of formal schooling, and that people's receptiveness to change is very intense at several times during adulthood. If we can take advantage of this aspect of human adaptability we will be meeting our own human as well as business needs.

What employers can look forward to from the learning cycle is continued productivity and protection of their investment. The investment of employees from age 22 to age 65 is an investment of between four and five million dollars to a company. The reason that educational technology is being considered as a way to implement the learning cycle concept is simply, "How do we protect and maintain that investment as an asset to the company in the most cost-effective way."

It has been estimated that corporations already spend over \$30 billion dollars per year on training, and employ 75,000 people full-time and another 75,000 part-time. The importance of that level of investment is clear and its efficiency will have to increase to meet projected demands.

To be more specific, the following are the five changes now occurring in industry that are relevant to the presentation:

1. The amount and complexity of job-specific training needed to work in modern industry has increased.
2. The need for continual upgrading of work skills has increased.
3. The number of people per year needing training has increased.
4. The geographical dispersion of a modern company's workforce has increased.
5. The cost of delivering training via lectures is becoming prohibitive.

None of these changes alone points to a need for educational technology. But taken together, they argue strongly for decentralized training that is individualized and can meet the continuing needs of adult learners.

One solution to achieving decentralization is to employ the power of today's educational technologies. Experience at DIGITAL has shown that these technologies can be used to deliver instruction cost effectively at a large number of widely dispersed locations, even if only a few people need to be trained at each location.

The issues forced on us by the changing nature of American industry are sufficiently urgent, and the use of educational technology sufficiently promising, that industry will be in the forefront of the widespread use of educational technology. Industry has a unique combination of resources and motivation to apply technological solutions to teaching on a large scale. Our traditional close partnership with colleges and universities will be of value to both higher education and industry as we continue to make great strides in the application of technology to education.

Industry is in the midst of decentralizing its workforce while traditional educational institutions remain largely centralized. A major difference today is that industry itself must provide much of the training to its employees because of increased specialization and technological change. And the median age of the population is no longer 16, but rather 32. This means that educational changes that influence large percentages of the population will have to occur beyond the years of traditional schooling.

Thus industry is in a position of needing a decentralized approach to training and also having to provide that approach itself. In many cases education and industry together can cooperatively address this lifelong learning challenge. Opportunities exist for using the resources of both institutions to address one of our major challenges - employee obsolescence. DIGITAL trains people which have diverse backgrounds and learning goals. Some join DIGITAL with significant experience while others have no job-specific experience. Some study courses to obtain detailed job skills, while others look only for generalities. Adaptive training programs should handle each learner slightly differently to maximize each person's performance against their own unique goals. In addition to traditional classroom instruction and self-paced

printed materials, DIGITAL presently use three classes of technology to achieve their training goals: linear video, computer assisted instruction, and interactive video.

Linear video is a very useful and easily decentralized training technology. It is helpful in sales training, in pre-sales customer training, and in conveying general concepts. This type of media presentation is supplemented with printed material. However, the type of presentation is limited and the level of interaction between the student and the technology is low. In many training situations we need a way to deliver decentralized training that is more adaptive.

Much of this adaptability can be provided by computer assisted instruction, which can be highly interactive and tailored to a student's individual needs. In particular, a student using computer using computer assisted instruction can:

1. Receive instruction at a time convenient for the student.
2. Control the pace and order of the instruction.
3. Ask for help, advise or the definition of an unfamiliar term.
4. Get feedback on what parts of a course to review.

Computer Assisted Instruction has an important role in a technical training environment although with some limitations. The graphic quality of today's computer assisted instruction is excellent, and for many training applications this is sufficient. However, standard computer assisted instruction technology cannot reproduce photographs or live action sequences with the clarity of video presentation.

Much of the material presented at DIGITAL requires video quality images. To satisfy this need while maintaining the level of adaptability provided by computer assisted instruction DIGITAL has begun to produce courses that take advantage of their own interactive video technology.

DIGITAL has developed an interactive video system that combines the best qualities of both computer generated and video displays. This system fully integrates video and computer assisted instruction by allowing both images to be displayed on the same screen. In addition to allowing two separate "windows" into the course, this allows video images to be overlaid with computer generated graphics and text. When the video source used is a videodisc player capable of randomly accessing 54,000 separate images the adaptability of this system far exceeds that of either linear video or computer assisted instruction on its own.

A student might be introduced to a new piece of equipment with a video sequence. If later evaluation shows that the student does not understand how a particular component works, part of the video sequence. If later evaluation shows that the student does not understand how a particular component works, part of the video sequence might be repeated with the component highlighted via computer graphics overlays. Or a student can be taught the assembly and disassembly of a complicated piece of equipment by allowing them to simulate the steps interactively, thus eliminating some of the need to maintain costly equipment at training sites. These techniques and many other variations allow

interactive video systems to be extremely adaptable to a large number of student backgrounds, abilities, and learning styles.

An example of the advantages of decentralized training using the adaptability of interactive video system is illustrated by one of DIGITAL's major groups. The total cost of training for this group will rise by over 40% during the next five years if that training is delivered in the traditional way, by lectures and self paced courses at training centers. These costs include not only the cost of producing courses but also the travel and time costs associated with training centers that are remote from many work sites. Using a model where much of this training is provided by interactive video systems installed in the group's field offices, almost all of that increased training cost can be saved. This results is consistent with the general trend toward higher costs for traditional educational delivery, which is labor intensive, versus the use of educational technology, which continues to benefit from decreased costs, especially hardware.

In addition to the cost savings, studies at DIGITAL of the initial use of interactive video courses have shown that properly designed interactive video courses can be as effective as lectures or self-paced courses for specific kinds of training. We have also found that the interactive video courses are excellent reference sources that allow employees to refresh their skills in particular areas just when they will be needed.

With all the benefits there are some limitations. The primary limitation with the creation of new hardware technologies the limited availability of software support. Many of the problems found with using educational technology are raised by the technology itself. In fact, educational creativity is outpacing the ability to implement instructional designs that are exceedingly complex. One of the promising developments in this regard is the use of artificial intelligence techniques in educational technology. Recent developments in both hardware and software have now made artificial intelligence techniques practical as an instructional tool. The promise of using artificial intelligence is that teaching systems can be built with all the technical knowledge of subject matter experts and all the teaching knowledge of psychologists and educators.

In summary, the major problem facing industry is change; employees last longer than the need for their present skills. The concept of the learning cycle in which a person's working lifetime consists of alternating periods of productive work and retraining, using some examples of specific changes in industry that required training to be decentralized and individualized. This combination of requirements makes educational technology a viable solution to training problems. Industry has a significant need to adopt educational technology on a large scale. The traditionally close partnership between industry and education in this country will be of great importance.

LA12/LA100 FUNCTIONS AND FEATURES

John L. Davis
Digital Equipment Corporation
Maynard, MA

E. Jory
Digital Equipment Corporation
Maynard, MA

Reported by Frank Fluke, DECUS Scribe Service

(Editor's Note: This material resulted from the "Scribe" service used at the St. Louis symposium. Look for more of this helpful service at Las Vegas.)

This article is a discussion of the LA12 and LA100, which covered the construction of characters using dot matrix techniques. Particular emphasis was placed on what can be done on the LA100 and the construction of personal fonts, binary image mode graphics or pixels, and new technical aspects of the LA12 and LA100 products. The first part of the discussion centered on dot matrix techniques for the LA100. These techniques consist of two parts, the decomposition of characters into small dots to get the image and the printing of those images.

The next entity introduced was the character cell, including its placement and definition. The size of a character cell is called the pitch. Pitch refers to both horizontal and vertical character size and is often confused with spacing. Resolution is the distance between possible dot positions within the character cell. For example, resolution influences how well diagonals can be drawn and whether they look like straight lines or steps. The LA100 has a resolution which is 1/14 of the cell size in the vertical direction. The LND1 has a resolution of 1/30. The VT100s is 1/10. When discussing printer resolutions you are talking about much greater resolutions because of the importance of hardcopy clarity which is often used for public display as opposed to video output. In the horizontal direction, the LA100 (1/33) has a finer resolution than the LND1 (1/30), while the VT100 remains at 1/10.

The next important parameter is dot size. On the LA100 the dot size is .13 mil. On the LND1 it is 3 1/3 mil. The dot size has two influences:

1. The finer the dots, the more detail that can be shown.
2. The finer the dots, the more dots that are needed.

Certain printer limitations must be faced, since the printer which puts the most dots on paper most augments the stress. It also increases the load on the power supply, heat dissipation requirements, storage requirements for additional features, CPU requirements (to process all the dots), repetition rate for

mechanical devices, and how fast the laser beam in the should be shut on and off. Thus, a compromise must be found between resolution, dot size, and the stress put on the printer.

One of the ways to save storage space is to avoid storing the white space which is at the right or at the bottom of the character. This is not always suitable for situations where extremely high resolution is required, since the quality of the output can be adversely affected. Another way to reduce storage requirements is to use algorithms which reduce the number of dots needed to form a character. For instance, the number of necessary dots can be cut in half by leaving a space to the right of each dot instead of having two dots adjacent to each other. This would not affect vertical quality, diagonals, or curves yet it cuts the stress on the printer in half, thus allowing speed to double. This can be carried one step further for even less quality and more speed. For medium resolution, start dropping dots in the vertical direction. Although this which does affect diagonals and curves, it permits a further doubling of speed. The purpose of all this is to open up an infinite number of characters.

This brings up the notion of a Dot Pattern Set (DPS). The DPS consists of a self-contained part (94 characters plus the auxiliaries) and a secondary part which contains a list of differences to be applied to the primary. In the DPS are stored all character patterns, all character set tables, all the character set selection data, and all supported attributes. This allows support of any new fonts. When a new character set is created, its character set table is placed in the DPS. Therefore, any existing machine is able to support it. Dot patterns can be modified in two ways:

1. New patterns can be defined(gives best quality).
 2. Space saving algorithms can be used (saves on number of designs needed).
- Q. What are the differences between the RA versions and the PC versions?
- A. There are a lot of misconceptions going around. The differences between them are as follows:
1. We changed the code on the PC to 8-bit code. The standard code that we use today is a subset, so if you have either one of those and use 7-bit code there should be absolutely no change. The only place that should show a difference is in Europe.
 2. The second difference is in the graphics aspect ratios. In the PC version we have included a choice of, I think, ten different graphics aspect ratios. The only one of our personal computers that wasn't quite right was the professional. It works, it prints, but circles came out a little flat and squares came out a little rectangular. The PC version corrects that.

3. The third difference is the EMIRFI standards. The RA version is FCC Class A and the PC is FCC Class B which is more stringent. Of the changes that can be made, the ROMs are interchangeable. The reason that people tell you that you can't make it into a PC is because you still haven't made it FCC Class B. You can make it work the same but its not FCC Class B because that takes some power supply changes and that's a little more extensive.

Concerning fonts, we are going to make custom fonts which I believe is the most cost-effective way to do it. The turnaround time is fast but we cannot be in the business of creating ones and twos for everybody. For someone with a little bit of volume we can do it. For people with any custom fonts or symbol sets, we should have the capability to make those for you within the next month or two.

Some terms must be defined before discussing binary image graphics. A pixel is defined as the smallest displayable dot. It is a spot of light on the screen and a blob of ink on the printer. The grid is the positions where you can place the pixels. On the printer we talk about dots per inch and on the video we talk about dots per screen. Resolution (previously defined) is understandable in hardcopy devices but is not understandable in video screens. Aspect ratio is the ratio between width and height. Overlap (when the spot size is significantly larger than the pixel) does not happen on the video screen because the dot on the screen is the same size as the pixel. Protocol is a way of encoding.

Some key characteristics have come into play in the graphics area and they turn out to be 'gotchas' when it comes to drawing things, so the differences need to be understood. With pixels, in the case of the printer, there exists a real problem. On DIGITAL terminals, the size of the wire can't be changed very easily on the fly, so it's pretty much constrained. The pixel is defined by the size of the wire and the amount of ink that gets put on paper. In the video, the pixel size is relative to the screen so if there is a large screen there is a large pixel. Horizontal resolution is determined by physical measurement and on the video it is a simple electron beam. The vertical resolution is in fact fixed by the vertical step (How much can the stepper motor be moved?), and that is 1/48" which turns out to be 72 dots per inch. There is an overlap between lines in that if 72 dots per inch is divided out, it is slightly smaller than the wire diameter. So there is a little bit of an overlap. There are some accuracy constraints from how the paper is moved around and the video has none of these problems because it is all relative to the screen. These are all key constraints.

In developing the protocol, it was to be; compatible with supported standards, compatible with the syntax, and able to pass information. The Device Control String (DCS) was chosen and the syntax was filled inside the DCS. This simply outlines what cannot be done. The printable data portion is a simple transformation where the received character code is transformed into binary data. The leading parameter in the DCS is used to define the aspect ratio. That is a key piece of information when interpreting pixels or any image like them. There are some aspect ratios that go from 5 to 1 down to 1 to 1. Pixels, then, are nothing more than a way of passing around a nice, clean image. This is great from the printer side because no bit map is needed.

Q. I have an LA12 and I tried to print something off of my VT125 with it, but I couldn't find a way to tell the printer using the keyboard to print out to the printer port. How can this be done?

A. The VT125 option plugs into the serial line of a VT100. There is no way, if you plug into a serial line, to get to a key and redefine it to be "print the screen". What they have provided is a kind of kludge mechanism so it can be done. What you do is simply enter the command screen to print the screen in your answer-back message, and then hit control-break whenever you want to print the screen image and it will come out. That's the hack.

Q. But only graphics?

A. Right, it only prints the graphics portion.

The LA100 is very flexible and can adapt to almost any kind of environment. It can be desk-top or stand-mounted, fitted with an acoustic cover to reduce noise, and fed using rear or bottom tractors or with roll paper in the back. The power supply is an international power supply that can be run between 85 and 250 volts, and between 43 and 67 hertz. The LA100 is Class B certified so it can be used in the home.

Communications are also flexible. The LA100 can be connected to the host using any EIA connection, a 20 mA connection, a modem, or the parallel interface. It has 14 stretch speed, 8 split speed, 8 character format, and can handle any kind of stop bit. The transmittal rate is limited to 60 CPS so it will not overload any host. It also has programmable answer-back, enabling or disabling of automatic answer-back, and coded disconnect. It supports XON/XOFF and busy. It has an input buffer which goes between 400 and 4K and supports long and short breaks for communication ending in a manner which is compatible with both U.S. and European requirements.

Flexibility is also found in form handling where the LA12 and the LA100 are much improved over the LA34 and the LA120.

Printing flexibility allows four print modes: draft, memo, letter, and graphic. There are 8 horizontal pitches in draft, 2x2 horizontal pitches in letter, and 8 graphic pitches. It has an infinite number of print styles and character sets. There are 270 resident characters and 180 characters either resident or plugged in, for a total of 450. The style as well as the quality may be selected either from the host or from the front panel. To improve speed, the LA100 handles auto-underline and does underlining on the fly.

There is flexibility in the way it handles the keyboard. The LA100 comes in both models RO and KSR. In KSR you can have: auto-repeat or not, keyclick or not, 8 keyboard configurations, disabled break key or not, and the RETURN can be a standard carriage return or a new line command.

The LA100 can be used as a line printer, a word processor, a standard console, a graphic printer, or a personal computer, each with many of the features adaptable to the chosen environment. It is compatible with the LA120, LA34, LA12, and the LA15, and feeds on the DECmate, PC, VT100, and the VT125. It works under both RSTS and VMS.

Some of the changes in the new generation - LA12 and LA100 - were stressed, such as form handling and the ability to handle the new DIGITAL 8-bit protocol which mainly doubles the extension of a repertory of controls and extends the number of characters that can be accessed with a single code by another 94. This reduces the data flow and increases the throughput of the system.

In order to set up the LA12, do the following:

1. Type the first character of the major piece, like "C" for Communications, etc.
2. Type the first character of the subcommand that is desired to understand or change. If you hit RETURN, it will tell you what can be done with that command, what the variations are, and then it will tell you the way it is set at the moment.
3. The last thing is the dialer, and if you type "D" you can start entering the numbers. The essential thing about the dialer is that it gives auto-dialing capability and storage for up to 26 numbers. The amount of characters per number varies depending on the numbers that you have. There is a fixed limit of 128 characters and there is an extra character needed for every number that you want to put in there.

The other things added in the version 2 code for the LA12 are: some bold printing capability, the variable aspect ratio, graphics to be compatible with the PCs, and an updated APL character set.

Keyboard dialing is actually very easy to do. To edit the keys a little, a simple editor is provided along with the facility to enter the digits, some separators so it looks pretty, and some wait characters for secondary dial tone detects. The keyboard dialer is pulse dial only and we did that to provide the most universal base thought possible.

Q. Is that a new ROM set?

A. The keyboard dialer is combination logic board and new ROM set. It was not the base board, it was the modem board we had to go in and change to put in the dialer circuitry.

Q. When is that coming out on the LA12?

A. About the first of July. The way the dialer works is that any time you are online and not in set-up mode, if you type "D", you go into dialer mode and it prompts and says, "You're in dialer mode. What do you want to do?" It's that simple.

DECUS 12 BIT NEWS

NUMBER 43

SEPTEMBER 1983

Contributions and correspondence should be sent to:

Robert Hassinger, 12 Bit Coordinator
c/o DECUS MR2-3/E55 ..or.. Liberty Mutual Research Center
One Iron Way 71 Frankland Road
Marlboro, MA 01752 Hopkinton, MA 01748

DECUS/Europe contributions are solicited through:

Lars Palmer
DECUS/Europe 12 Bit News Liaison
Hassle
Fack
S-431 20 MOLNDAL 1
SWEDEEN

(Please include referent to Newsletter number and page when inquiring about material published).

NEWSLETTER SUBMISSIONS

Submissions are accepted at all times and are normally used int the next issue to go to press regardless of date of receipt.

Material submitted in machine readable form is particularly desirable because it can be edited and incorporated into the newsletter format more easily. Higher quality reproduction is also possible this way. Contact Bob Hassinger for further details on acceptable media and formats if you plan to make a submission in machine reacable form.

IN THIS ISSUE

I got word this issue was due to go in less than a week before the deadline. This was the first had heard about publication plans since the last issue went in. As a result and due to the short time since the last issue reached you, the number of items is less than normal but they are interesting.

There are reports on 12 bit topics fromt he Spring Symposium, a report and commentary on the future availability of OS/278 for the DECmate II, an update on PASCAL-S and some news on current goings on from Wally Kalinowski and Louis Tribble. There is also a proposal from Wally and Louis about the future of 12 bit software with an invitation for your input - please respond.

PAGE 113

12 Bit Coordinators

Robert Hassinger
- address above -
(617) 435-3452

COS/DIBOL and WPS Liaison

Lawrence H. Eisenberg
17141 Nance Street
Encino, California 91316
(213) 788-0354

Education, Multiuser systems, PASCAL

Father Geoffrey Chase O.S.B.
Portsmouth Abbey School
Portsmouth, RI 02871
(401) 683-2000

Representative to DECUS Product Planning Committee

Jim van Zee
Lab Data Systems
10320 Ravenna Ave. NE
Seattle, Washington 98125
(206) 522-6950

FUTURE DEADLINES

The publishing plans for this new newsletter format are still in a state of flux. At this point it looks like I will be closing the next issues about the first of October and early in December. I know this will leave very little time after each issue arrives but the new system has added layers of overhead that make newsletter turnaround time longer than we used to have. Contact me if you have problems or questions about deadlines.

PASCAL-S

The latest word is that PASCAL-S is still going strong. I believe the original authors are favorable to the English version being submitted to DECUS. At last report the DECUS submission was contemplated in the near future. More next time.

SPRING DECUS SYMPOSIUM

One of the most interesting things at the Spring Symposium in St. Louis from the 12 Bit point of view was the news that the DECmate II was doing better than anyone had predicted. It seems that sales have exceeded all projections and the extra income is helping encourage a number of related developments that had been in the it-would-be-nice-if category up until now.

In the machine room we saw three DECmate IIs with 10 Megabyte Winchester disks (rather than the 5 Megabyte version that has been available for the other DEC personnel computer lines) and it was inside the CPU box rather than outboard as had been discussed earlier. We also saw a color monitor connected to a video board that does the same type of thing as the bit map graphics on the other personnel computers. This had not been a part of the DECmate II offering. Both these items were there and working for

PAGE 114

those who noticed them even though they had not been announced as far as I know.

The graphics board was running a series of demo programs that were said to actually be running under OS/278 - a very likely story since it would be hard to get the performance we saw any other way - (see notes on this subject later).

There was a strange hybrid system on the Winchester disks. The physical disk was broken into several logical disks. When you booted the system you came up in CP/M but if you knew the right magic you could boot one of the other logical disks and you found yourself in a somewhat unfamiliar version of OS/278 (the prompt character has been changed for example - why? the period has been good enough for the last 14 years and it is still good enough for RT-11 which copied it from OS/8). There was this little problem that if you got the magic wrong OS/278 was blown away, never to come back - by the end of the Symposium all the copies of OS/278 were gone as a result. This is not the way OS/278 would actually work on the DECmate II. You would never be required to have the CP/M option to get to OS/278. It was just an artifact of the unofficial demo system.

We had some interesting papers on PDP-8 base applications such as a news wire system with very stringent reliability and recover requirements. I hope at future Symposia we can get back to more user presentations like we used to have years ago.

We anticipate a session at the Fall Symposium giving an introduction to the OS/8 family. I hope that at the Spring Symposium more of us will be able to attend and we can have a more extensive program. Get busy now thinking about what you can do. Individual papers, user panels, workshop sessions, anything you think would be good. Contact me if you want help - the call for papers for the Spring Symposium will be out and due back very soon now! We have to get busy.

"DECmart"

During the St. Louis Symposium I was introduced to the latest thing in DEC marketing for PDP-8 systems and options - "DECmart". It is a "Computerized Shopping Service" that you dial into with a terminal. The system can give you descriptions, prices and availability, take orders, let you request a written quote or a call back and you can add your name to a mailing list. Both new and factory refurbished equipment is listed. The emphasis is on PDP-8 items but there are some non-PDP-8 items too. I found a good bit of useful information in the data base.

The system belongs to DEC's Traditional Products Group. I was told it is running on a PDP-11/60. I suspect that if it works out well for them, TP&G might expand the system to cover some of their other products lines.

To use the system, dial 603-884-4427 and connect your terminal through your modem. There is no logon required - the system just takes over and tells you what to do. Give it a try, you might find an interesting bargain or some helpful information.

OS/8 FAMILY RUMORS, REPORTS AND THOUGHTS

When the new DECmate II was announced at the Spring DECUS Symposium a year ago, there were no plans to make OS/8 family (OS/8 - OS/78 - OS/278 - etc.) software available for it. When I got up and asked about it at the Symposium, everyone looked at me as if I was from Mars. No, DEC had no plans then or in the foreseeable future. Why would anyone want OS/8 anyway?

Well, as we well know, there are a number of good reasons why one might want OS/8 on the DECmate II. When I looked into the subject further, I found that among the first things that had been done during the DECmate II software development was to get OS/278 working on it. This was needed in order to work on WPS and COS since the OS/8 family is the only available support for general purpose and assembly language programming of the 12 bit systems so DEC supports and develops the code for COS and WPS with the OS/8 family systems.

Finally, at the Spring Symposium this year, we heard a new tune from the DEC people. They had seen the light and were planning to release an improved version of OS/278 for the DECmate II. The only question was a decision on how extensive a project they wanted. How much new work on documentation to do and so on. The decision was to be made "within the next few weeks" and the product was to be ready in the Fall. Past experience made me worry a little about the sound of that but it was a whole lot better than anything before.

Since then a few things have come to light. It seems a lot of the package is working, including an upgraded version of BASIC. Unfortunately, at last report FORTRAN IV had not been patched to work on the DECmate II and it was not clear that anyone planned to include it in the new release - a great oversight I think. Worse yet, the new BASIC was reported to be working only on the DECmates and it was not clear that it would even run on the DECmate I when it was finally released. I am not clear about the status of other software or the operating system proper but it sounds like it is quite possible that there will be no compatibility with older hardware versions.

This is very disappointing! It seems the people making 12 bit decisions now at DEC have lost sight of the guiding principles of the past that helped make DEC's 12 bit systems and the OS/8 family the great success they have been. In the past the principle OS/8 development people were always firmly committed to new software being compatible across all the the older systems. For example, when you run FORTRAN IV it checks the hardware and dynamically reconfigures to work with what it finds. This was not all that difficult to do technically, primarily it took a desire and commitment to being sure it was done. I think history has proven the wisdom of the pioneers who maintained the commitment for so long. The unusually strong compatibility from machine to machine and release to release did a great deal to popularize the 12 bit systems. Customers liked to know their software investment was protected. Lets hope today's developers and managers learn these lessons before it is too late.

STOP THE PRESSES - on deadline day for this newsletter as I am about to mail it off, I now get new reports from my contacts to the effect that the OS/278 project has run into funding problems and is on "hold". Lets hope

that this turns around and DEC gets on with the project, much of the software is far enough along for field testing, how much more could be needed to finish the job? Can't a multi-billion dollar company, second only to IBM, find a few dollars to enhance one of their most successful new products?

The DECmate II has been reported to be one of the few bright spots in a generally dismal sales year for DEC. Availability of OS/278 for it would further encourage sales. Using OS/278 avoids the need to buy a CP/M card and software for the DECmate II and it saves the slot that the board uses. It will not be long till the number of option slots in this machine will be a problem. Saving one will be increasingly important as more options become available.

There were a number of people at the Symposium this Spring concerned with the fact that they had OS/8 based software packages that they wanted to be able to continue selling. They need OS/278 on the DECmate II to do this. When they sell their packages they are helping boost DEC sales too.

There is also seems to be a more subtle point in favor of OS/8 for anyone interested in the details of how I/O is programmed at the machine level in this machine. For example, I want to interface a DECmate II to a digitizing tablet so that a lab that uses the system for word processing can also do a little data collection from time to time. This type of thing requires very close coordination of the I/O programming to be successful. The required control is not available under COS and WPS can not do the job at all. If I try to do the job with what DEC gives me with the CP/M option I will have to do use a CP/M BASIC program. This does not give the needed control either. Even if I go out and buy additional software to do assembly language programming of the CP/M processor I will find that I still cannot get the control of the I/O that is required.

The reason is that that the CP/M processor does not actually do any I/O

with special software the user has no access to. The only I/O interface available to the CP/M programmer is at the level of CP/M's version of a device driver (and I have not even seen any documentation of that interface yet) - a line or buffer at a time. You can not use interrupts and all the rest of the facilities you need to really control peripherals and I/O. You can not even sit in a tight loop waiting for a each character to be input - a common and customary practice on small single users systems.

To do these things, you have to program the 12 bit processor directly and the only way to do that is in an OS/8 program written and run under the OS/8 family of operating systems and support software. Indeed, if I wanted to I could avoid writing an assembly language program and do my data tablet application under interrupts directly from an OS/8 FORTRAN IV program with all the control I need. The hooks and tools are there to do it. Applications that are tightly integrated with peripheral devices can benefit greatly from access to the OS/8.

What if I want to drive a particular device, perhaps a printer that has different characteristics for example? In OS/8 I just write or modify a device driver to do what I need. What do I do in COS or WPS or CP/M? Forget about it - that's what I do! I am stuck with what DEC gives me and

that tends be limited to support for current DEC hardware - take the LQP printers for example.

While the majority of DEC's customers are not going to get involved in this level of programming, having it available to the people who write software packages means it is possible to build and market very attractive programs that extended the usefulness of the machines and encourage sales. I know of a number of people who would like to market software products based on the DECmate II because it is such a good machine at the right price. Their efforts get an added marketing advantage from the fact that when you buy the DECmate II to run their package you can also get one of the top professional word processing system at the same time - not many other small systems can match that advantage. The key to availability of many of the software products I have heard about is the availability of OS/278 on the DECmate.

On the other hand - one hopeful thought - if DEC is not going to release OS/278 for the DECmate II, maybe someone will finally decide to release it to DECUS so the users can have the chance to maintain and enhance it. We (the 12 bit users) have asked many times over the years for this. When a software product is removed from the catalogs it should be released with sources to DECUS so it will continue to be available and can be maintained by those who still need it. There are many examples - MACREL, RTS-8, OS/8 V3D (DEC is working on the fifth version of OS/78 and OS/278 since the last release of OS/8 five years ago but people with PDP-8s still need access to it). As it is, the older material has been getting lost too often. I get many calls from people who need some piece of DEC software and find that despite any policy or procedure that says otherwise, they simply can not get it from DEC. In effect it is just gone and they have no where to go. Releasing the old software to DECUS would help the users as well as DEC.

IRONS IN THE FIRE

From: Wally Kalinowski and Louis Tribble

Eugene Lynch has just submitted versions two and three of his TKPLOT program to DECUS. TKPLOT replaces PLOT in the F4 plotting routines, allowing plots to be displayed on a Tektronix 4010 terminal and stored to disk for later plotting. A routine to execute stored plots is included. The revision was inspired by the recent availability of terminals suitable for use both as a console device and a 4010 (the original package assumed that the 4010 was not the console device). While at it, he added features for interactive programming.

Dr. Lynch is now working on implementing the symbiont for OS8, which will allow printing, plotting, and other tasks to run in the background without tying up the computer. The problems at Eckrohrkessel mentioned in the previous newsletter may soon be solved.

We came by a public domain copy of Spice, a large Fortran program from UC Berkeley which does electronic circuit analyses. If anybody out there has time to implement this on the 8, let us know.

We have been working on a reasonably elaborate macro library, etc. for Macrel and are interested in anything anyone else has done (we will try to contact Lars Palmer). Is there still any interest in a DECUS submission?

RE: CALL TO ARMS

From: Wally Kalinowski and Louis Tribble

We have decided to break the first rule of blissful ignorance: we volunteer. The consensus among the people we keep in touch with is that there is no more money to be made selling PDP8 software. But there exist many really useful - even exciting - programs that are proprietary or haven't been submitted to the library. We think that if all interested parties (1600 people are on the 12 Bit Sig mailing list) kicked in ten or more dollars, we (DECUS) could acquire the rights to many of these programs and distribute them. Further, the DECUS library seems to be approaching critical mass, and something ought to be done about that. Hence, our proposal:

We are willing to do the necessary negotiating with the authors/owners of all the significant programs we can find. We will combine these with the current DECUS library (note 1) into a neat package (including all printed material) that is ready for distribution (note 2).

We have already arranged for some programs. Bill Haygood has offered sources for MULTOS8, a multiuser OS8 with background printing. Jim Van Zee has offered his wonderful high-density, high speed floppy handlers. We have a file transfer program that reads/writes ASCII or binary files of any size from/to a disk at up to 9600 baud. And, probably, Father Chase's revision of the Pascal S compiler will be available. See our other note for more programs that may be ready by the time this gets rolling. Programs we know of (most are still proprietary) include very nice editors, cross assemblers for various micros, a very good text formatter, Forth, a superior Basic, LDFocal, several choice handlers, a full Pascal compiler, and lots more.

What interested readers need to do is return the addressed questionnaire. If the response is adequate to justify the effort, we will collect money, set (by survey?) priorities, acquire software, and start assembling the package. If anyone knows of suitable software, please contact us. For example, who has copies, rights, or whatever for FILES and BATCHM?

Note 1: Weeding the Library

There are many programs in the library that are either redundant or obsolete. When we package the library, we want to leave them out. We will pass them on to other evaluators (e.g. someone using paper tape), or to the DECUS archives. At any rate, we would list our choices in the newsletter.

Note 2: Distribution

There are two possibilities: DECUS or non-DECUS. We hope for the former. Our activities should greatly simplify the distribution problem. We will package in units of floppies (filling each) and for each unit, package the printed documentation in one folder. The number of floppies (20?) would be far fewer than there are now, and for floppy n, a clerk need

only grab folder n. For tape or hard disk, one or more floppy sized units could be placed on each reel or platter. As for reproducing the library on a PDP11, we have a suggestion for floppies. In addition to their other advantages, Jim Van Zee's handlers use a PDP11-compatible format, which means disks so recorded can be directly duplicated by a PDP11. For other media, a PDP8 is necessary, but if most people can handle floppies, perhaps volunteers can perform the conversion?

If DECUS refuses to distribute, then we can copy off some complete sets and will help organize volunteers to copy more (of course, these volunteers can expect to be sent media before they send copies!).

FROM THE EDITOR

Welcome to the first newsletter of the DECUS Personal Computer Special Interest Group. I must confess that I'm in something of a dilemma as to where to begin. Since this is the first issue, I have no choice but to make some assumptions about who will be reading this issue, your level of exposure to the organized community of Digital Equipment Computer users which is DECUS, and even the extent to which such information is helpful.

The Personal Computer SIG was officially formed at the Spring DECUS symposium in St. Louis. Bob Branton agreed to serve as the interim SIG chairman, and over thirty people were involved in the working group which led to the SIG's formation.

The primary purpose of a Special Interest Group, within DECUS, is to serve as a focal point for activities and information concerning a particular product or application. SIGs sponsor newsletters, symposia sessions and Local Users' Groups. In addition, SIGs serve as an organized voice to Digital, and can be an effective means for letting Digital know what you like in their products, what you'd like to see changed, and directions you'd like them to take with future products.

The first fruits of the PC SIG are the sessions the SIG will be sponsoring at the fall symposium, October 24-28 in Las Vegas. In this you will find a list of these sessions. Stephen Finch is serving as symposium coordinator, and along with several others, put in a lot of effort to get this schedule together.

It is my feeling that Local Users' Groups are really the reason for the existence of a Personal Computer SIG. People using the Rainbow or DECmate for small business applications may not have the opportunity to attend a symposium half a continent away, and engage in the kind of personal information sharing that is the principal benefit of attendance. For them, the LUG is the only place to get it. LUG formation is one of the primary goals of the PC SIG.

The SIG newsletter, at its best, can serve as a communications channel between DECUS and individual users, and among the users themselves. It is my hope that you will use this letter to exchange ideas, to air and solve problems, to brag about successes and to share information.

In the future, I hope to include reports from the Fall session, information on the formation of Local Users' Groups, including both what groups are being formed and how to go about forming one, and especially letters and articles from people who are using the Digital Personal Computers. So let me hear from you, and let's get the sharing process started.

Contributions should be sent to:

Personal Computer SIG newsletter
Mark DeMoss, editor
Dallas Computer Works
2408 Texas Drive #109
Irving, TX 75062

PERSONAL COMPUTER SIG
Volume 1 Number 1

From the Editor 123
Fall Symposium Schedule 124
Library Reprot 129

***** MONDAY *****

Title: PERSONAL COMPUTER SIG BUSINESS MEETING
Length: 1 hour
Scheduled: Monday, October 24, 10:00-11:30 a.m.

Title: PERSONAL COMPUTER PRODUCT PANEL:
RAINBOW, DECIMATE II, AND PROFESSIONAL SERIES
Length: 1 1/2 hours
Scheduled: Monday, October 24, 1:30-3:00 p.m.

Title: A GUIDE TO PERSONAL COMPUTER DOCUMENTATION
Length: 1 hour
Scheduled: Monday, October 24, 3:00-4:00 p.m.

Title: TELEPHONE SUPPORT FOR DIGITAL'S PERSONAL COMPUTERS
Length: 1 hour
Scheduled: Monday, October 24, 4:00-5:00 p.m.

Title: SOFTWARE PRODUCER PROGRAMS
Length: 1 hour
Scheduled: Monday, October 24, 5:30-6:30 p.m.

Title: USING DIGITAL PERSONAL COMPUTERS WITH AN EXISTING
DIGITAL COMPUTER
Length: 1 hour
Scheduled: Monday, October 24, 6:30-7:30 p.m.

***** TUESDAY *****

Title: GRAPHICS ON THE RAINBOW-100
Length: 1 Hour
Scheduled: Tuesday, October 25, 8:30-9:30 a.m.

Title: COMMUNICATIONS ON THE RAINBOW 100
Length: 30 minutes
Scheduled: Tuesday, October 25, 9:30-10:00 a.m.

Title: CP/M 86/80 ARCHITECTURE AND INTERNALS
Length: 1 1/2 Hours
Scheduled: Tuesday, October 25, 10:00-11:30 a.m.

Title: STRATEGY FOR IMPLEMENTING PERSONAL COMPUTERS
IN A LARGE ORGANIZATION
Length: 1 hour
Scheduled: Tuesday, October 25, 1:30-2:30 p.m.

Title: RAINBOW 100 -- ALTERNATIVE OPERATING SYSTEMS:
A COMPARISON
Length: 1 hour
Scheduled: Tuesday, October 25, 3:00-4:00 p.m.

Title: RAINBOW 100 THIRD PARTY SOFTWARE FORUM
Length: 1 1/2 hours
Scheduled: Tuesday, October 25, 4:00-5:30 p.m.

Title: RAINBOW TECHNICAL Q & A
Length: 30 minutes
Scheduled: Tuesday, October 25, 6:30-7:00 p.m.

***** WEDNESDAY *****

Title: DECMATE II: A SMALL BUSINESS SYSTEM
Length: 1 hour
Scheduled: Wednesday, October 26, 8:30-9:30 a.m.

Title: DECMATE II COMMUNICATIONS
Length: 1 hour
Scheduled: Wednesday, October 26, 9:30-10:30 a.m.

Title: DECMATE II GRAPHICS
Length: 1 hour
Scheduled: Wednesday, October 26, 10:30-11:30 a.m.

Title: DECMATE II HARDWARE
Length: 1 hour
Scheduled: Wednesday, October 26, 12:30-1:30 p.m.

Title: DECMATE II: DATA BASE MANAGEMENT
Length: 1 hour
Scheduled: Wednesday, October 26, 2:30-3:30 p.m.

Title: DECMATE II SOFTWARE DEVELOPMENT TOOLS
Length: 1 hour
Scheduled: Wednesday, October 26, 3:30-4:30 p.m.

Title: P/OS STRUCTURE: AN INTRODUCTION
Length: 1 hour
Scheduled: Wednesday, October 26, 5:00-6:00 p.m.

Title: PERSONAL COMPUTERS IN HIGHER EDUCATION:
THREE CASE STUDIES
Length: 1 hour
Scheduled: Wednesday, October 26, 6:00-7:00 p.m.

***** THURSDAY *****

Title: PROSE-PLUS FOR THE PROFESSIONAL
Length: 30 minutes
Scheduled: Thursday, October 27, 8:30-9:00 a.m.

Title: P/OS GRAPHICS
Length: 1 hour
Scheduled: Thursday, October 27, 9:00-10:00 a.m.

Title: PROFESSIONAL 300 HARDWARE INTERNALS
Length: 1 hour
Scheduled: Thursday, October 27, 11:00-12:00 p.m.

Title: APPLICATION TUNING FOR P/OS
Length: 1 hour
Scheduled: Thursday, October 27, 2:30-3:30 p.m.

Title: P/OS INTERNALS FOR APPLICATION DEVELOPERS
Length: 1 1/2 hours
Scheduled: Thursday, October 27, 3:30-5:00 p.m.

Title: PROFESSIONAL 300 USER PANEL
Length: 1 1/2 hours
Scheduled: Thursday, October 27, 5:00-6:00 p.m.

***** FRIDAY *****

Title: THE PROFESSIONAL TOOL KIT: DEVELOPING SOFTWARE
FOR THE PROFESSIONAL 300 FAMILY Length: 1 1/2 hours
Scheduled: Friday, October 28, 8:30-10:00 a.m.

Title: PROFESSIONAL 300 THIRD PARTY SOFTWARE FORUM
Length: 1 1/2 hours
Scheduled: Friday, October 28, 10:00-11:30 a.m.

Title: ALTERNATIVES TO PO/S ON THE PROFESSIONAL
Length: 1 hour
Scheduled: Friday, October 28, 11:30-12:30 p.m.

Title: PROFESSIONAL 300 TECHNICAL Q & A AND WISH LIST
Length: 1 hour
Scheduled: Friday, October 28, 1:30-2:30 p.m.

Title: PERSONAL COMPUTER SIG WRAP-UP
Length: 1 hour
Scheduled: Friday, October 28, 2:30-3:30 p.m.

LIBRARY REPORT

The DECUS Library has received its first submission for the DEC Professional 350. The MCR, TECO, SRD and diskette COPY package was developed by Richard J. D. Kirkman of Filetab Support Services, London, England.

The COPY alone is worth its weight in gold, and that's only one quarter of what is offered. Why wait for a "native mode" kit, get these tools in hand now. For ordering information contact

DECUS
Order Processing
One Iron Way
MR02-1/C11
Marlboro, MA 01752

By the way, if you have submissions for the Rainbow, DECmate II or Professional series, write for a submittal form today (same address as above).

R. D. McKinney