



What We've Found

Listed below are some significant findings that we have confirmed through our testing:

- o The use of color not only enhances the appearance, but also the usability of an information product.
- o Providing "tutorial" style information vs. "reference" style information not only benefits inexperienced data processing users, but also experienced users.
- o Information products that contain a great deal of information are acceptable as long as the information is easy to locate.
- o Inexperienced data processing users generally have a high satisfaction level if they are able to complete a given task successfully regardless of the length of time it takes to complete it.
- o A high percentage of people primarily use examples to complete a given task, rather than reading text.
- o The cost savings of identifying and correcting problems while a product is still in development vs. out in the field is significant.

Some Final Comments

Testing information products has become an important part of our Information Development cycle. The techniques we use have undergone a great deal of change in an attempt to satisfy the end user's needs first and foremost, without jeopardizing information development cycles.

SHARE SESSION REPORT

61	A522	Process Control Graphics	15
<u>SHARE NO.</u>	<u>SESSION NO.</u>	<u>SESSION TITLE</u>	<u>ATTENDANCE</u>
Graphics Applications		Jim Boyle	BKC
<u>PROJECT</u>		<u>SESSION CHAIRMAN</u>	<u>INST. CODE</u>
Bendix-Kansas City Div, Box 1159, Kansas City, MD 64141 (816) 997-4773			
<u>SESSION CHAIRMAN'S COMPANY, ADDRESS, AND PHONE NUMBER</u>			

A Process Control Display Language

L. Hoffman
 J. M. Huber Corp.
 Thornall St.
 Edison, N.J. 08818
 201-549-8600

Installation Code HUB
 Graphics Applications Project

Session A522
 Tue. 4:30 P.M.
 Aug. 23, 1983

ABSTRACT

This paper presents facilities required by both plant operations and plant management. The designs are to be used as a yardstick to measure any process control graphics system.

This display language is designed for rapid response in a process control environment. The commands are extensive, yet simple, constructs for both pictorial and data displays of process and associated data.

The language is not designed for any particular computer or display device.

106

Tasks and Scenarios

The test coordinator develops all tasks and scenarios for the test subjects to perform. The tasks are developed about the same time that the writers begin writing.

The tasks and scenarios are written in a friendly tone in an effort to make the test subjects feel as comfortable as possible.

Measurement criteria is developed for each task. Length of time acceptable to perform the task and whether assistance is acceptable are some examples of measurement criteria that we might use for a task.

Test Lab Setup

To run an effective test, we make every attempt to set up our test lab like the intended user environment.

The test lab is divided into two rooms--the main area where the test subjects work, and the monitoring room where the monitors view the test subjects. The rooms are divided by one-way glass.

Test Subjects

The test subjects that we choose must match the audience that will be using the product.

Bringing in test subjects who match the intended audience is very important because the data that we obtain from our tests would be considered invalid if we chose the wrong test subjects. If, for example, the stated audience for a particular information product is novice users with little or no data processing experience and a high school education, we would not use individuals with college experience or a programming background. The individual with college experience would most likely have more knowledge than the novice user and could also make more assumptions and comparisons to other experiences. As a result, information that a novice user would need might be missing from the information product.

Team effort between test subject and test monitor is stressed throughout the usability test, especially at the beginning. As you would expect, people don't like to make errors, especially if someone is watching them. Therefore, we stress that the information product is being tested--not the test subject. We also stress the fact that we not only expect, but **want** the test subjects to find errors.

Documentation Used for the Test

The documentation that the test subjects use during a usability test resembles the final product as closely as possible. Retrieval can only be verified if the index, table of contents, and glossary are complete. If the final product will be a particular size and contain color, we make every effort to have the documentation cut to size with color pages. Physical characteristics play an important role in how usable an information product is. For example, if a quick reference is to be designed narrow enough to fit into a shirt pocket and printed horizontally instead of vertically, we would lose valuable data if we were to test it printed vertically on 8 1/2 x 11 paper. We would be unable to determine how the style of the quick reference affected the usability of the product.

Monitoring Techniques

In most cases, we have one monitor per test subject. The monitor is responsible for:

- o Providing tasks and scenarios to the test subject
- o Observing and documenting the progress of the test subject
- o Debriefing the test subject
- o Writing up the problems found during a test
- o Providing encouragement and assistance to the test subject

The test coordinator teaches the test monitor how to perform all of these responsibilities before the test begins.

Problem Reporting

The test coordinator is responsible for reporting the problems found during a usability test to the appropriate people. We assign a severity code to each problem--from high impact to recommendations.

Both the test coordinator and the test monitors participate in a daily problem writing session each day after the test subjects leave.

The problems are tracked and the problem fixes are verified by the test coordinator.

Recommendations and conclusions are provided at the end of the test.

A Process Control Display Language

L. Hoffman
 J. M. Huber Corp.
 Thornall St.
 Edison, N. J. 08818
 201-549-8600
 Installation Code HUB
 Graphics Applications Project
 Session A522
 Tue. 4:30 P.M.
 Aug. 23, 1983

TEXT AND DATA SOURCE:

Text should come from some source, such as the plant table (better known as a data-base). Most text is not generated, but comes from some pre-planned source.

CRT RESOLUTION:

The characters displayed should be available in different sizes and colors, and maybe even different orientations. Character resolution should be such that both textual material and ISA standard symbols are readable from at least 2 feet away from the face of the CRT. (Note: ISA is the Instrumentation Society of America. It is a "standards" group with many standard symbols for the instrumentation field.)

LINES AND ISA SYMBOLS:

Lines, shapes (process diagrams), and probably ISA standard symbols should be presented in any color. Lines should be displayed as solid or various types of broken lines. Lines cannot be placed on top of each other. Lines that intersect are different from lines that cross in the two-dimensional plane. This differentiation should be quite clear to prevent a mis-interpretation of the diagram.

COLORS FOR SYMBOLS AND VALUES:

Process values, including tank levels, flow rates (and appropriate labels and units) should be displayed in various colors and sizes.

STATUS DISPLAY:

Process status displays for both present, recent, and historical values should be presented in various colors. Most status displays will be bar charts or line graphs. A useful feature may be to provide a projection of future conditions, based on past history.

ABSTRACT:

This paper presents facilities required by both plant operations and plant management. The designs are to be used as a yardstick to measure any process control graphics system.

This display language is designed for rapid response in a process control environment. The commands are extensive, yet simple, constructs for both pictorial and data displays of process and associated data.

The language is not designed for any particular computer or display device.

PROCESS DISPLAY CHARACTERISTICS:

There are some characteristics that are required for a process graphic display.

The following considerations are not an exhaustive list, but are special considerations for a "host-based" process control system. These considerations are based on problems and requests from our process control users in the field.

ALL displayed values will come from the plant table, or data-base. There may be a need to reference the data-base for set-up of a display, especially in the case of direct input. (This way, the description and units of a displayed value can be presented so that nothing will over-write that text.)

The description of physical units, which should be displayed after a value, should be obtained from the plant table since this is the master description. The values are calculated by applying calibration factors such that these physical units result.

By the use of the stored units description for values, there is little or no room for error in any displayed value.

SPECIAL FEATURES:

There are features that should be specified for process displays. Some of these exist in other vendor software, and some are things that we feel should be included in the capabilities.

There should be more than a "static" display of the plant, or section thereof. The displayed values may change color, depending upon tolerances or some other criterion. Not only would we get a value displayed, but we would have an idea of its relation to some critical value.

If a tank is displayed, the pictorial level in the tank should be displayed. This is quite prevalent in other vendor displays.

The status of an element should be displayed as a different symbol or color. This would allow the operator to differentiate between an open or closed valve, or a motor that was started or stopped, etc.

An element which is in an "out of limit" condition should be displayed in an "alarm" color, or flashing, to catch the operator's eye. Most examples I have seen turn either a value or a process symbol to a RED color when the element is in an "alarm" condition.

Process variables, once relations are established, could be displayed in a number of ways.

Trend displays could have the option of not only displaying the mean value, but also could display the standard deviation and the mean value.

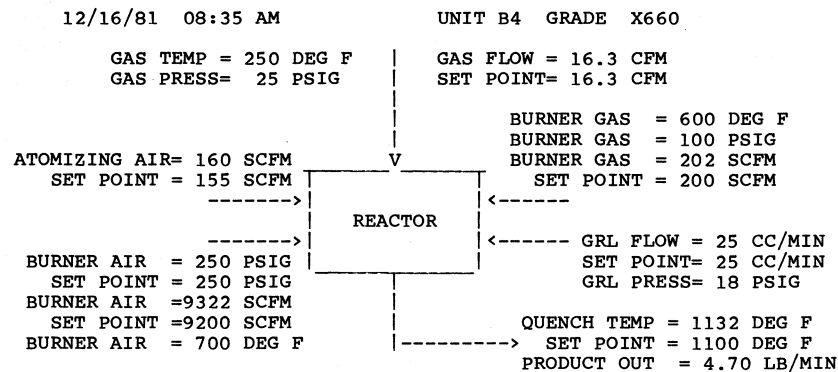
There is a definite need for the ability to display how far the

process is from optimum or set point.

A "pie" chart display may be useful for control loops. This may be a display of the degree of control in the loop, or the percent of full scale for the loop.

A "bar chart" of loop values with a range from 0 to 100% may be useful in determining where the loop control points are. The operators could tell that a loop is near the end of control by how close the values are to either end of the scale.

KIVIAT charts are used in computer systems for resource control. Many variables are presented on a polar coordinate system and normalized to a scale of 0 to 1. A properly tuned system is represented by a circle. This same technique may indicate a "well-tuned" plant that is producing product in an optimal fashion. This would be an area for future research.

SAMPLE OF A GRAPHIC DISPLAY FOR A TYPICAL PLANT

109

THE COMMAND LANGUAGE CONCEPT:

This method has a set of commands that would build a display through a software interpreter. The commands could be entered into the system by any editor and then the display could be generated at some later time or on demand.

This method would be similar to the process control system language, in that the commands would be compiled into some internal representation and then interpreted at high speed. The translation into machine pseudo-code reduces the repetitive character instructions that would otherwise be done to recognize english commands.

We have seen one rather popular system which uses the command language concept. The capabilities and speed are such that this is a very good way to generate, display, and modify graphics for a process control system. More about this language later.

ADVANTAGES:

This method is not dependent upon a particular CRT.

The commands can be entered into the system at any time in a "background" fashion.

Special routines (macros) can be set up for any shape. Since these are combinations of basic commands, the routines do not have to be set up by a programmer.

The display could be presented on a line printer for "off-line" debugging.

Standards for symbols, connections, shapes, etc. could be enforced since a program does the translation from some "english" command to the graphic commands.

If a "standards" change were necessary, it would be much easier to change the translator than to change all commands to reflect a new standard, considering the investment in time and effort on existing displays.

Special plot commands could be entered into the command stream. If some special function were purchased for a plotter, the functions could be checked-out without any system changes. The translator would pass the special (or "native mode") commands directly to the plotter.

The display could be set up anyplace, even some remote location.

DISADVANTAGES:

The display is not visible while the commands are being entered. (It may be possible to build an interpreter into an editor so that each command could be seen on top of the text being prepared.)

The display will probably not be correct the first (or second) time the commands are interpreted. The number of iterations will be a function of the capabilities of the display designer.

A different interpreter is required for each different type of CRT or printer.

An interface program to the process control system would have to be developed to display the updated values from the plant table.

There is a question of lines that cross. How does a general display differentiate between two lines that might fall on top of each other? (It is presently left to the human display designer.)

How does the display denote an intersection of two lines versus two lines that merely cross on the display? (This is presently left to the human display designer.)

A COMMON INTERFACE?

One of the above disadvantages is that one cannot see the result of a command until all commands have been entered, compiled, and run. This is very much like "batch" programming which has been done for 25 or 30 years. Only one error is generally found in each trial run!

Some CRTs allow the operator to build a display on the face of the CRT and then transfer the commands (stored in the internals of the CRT) to a host computer. The problem with this concept is that the commands for one brand of CRT are not the same as the commands for some other brand of CRT. Thus, there is no portability across devices.

If we could translate the commands for one brand of CRT into the language presented here, we could then run the translated commands on any other CRT!

This language could be a useful portability tool and allow an easy method for graphics entry.

For example, the HMW color CRT allows an operator to draw lines, figures, circles, ISA symbols, and text. This information is in the display internal buffer. When the "page xmit" key is pressed, the entire buffer, which represents the display in some numeric form, is sent to the computer. A program can be written which will recognize all of the forms which have been displayed on the CRT. Lines, text, circles, ISA symbols, etc. would then be translated into the graphics language of this paper.

Since the original HMW display is now in a general form, the display could then be presented on any CRT supported by the system.

To the best of this author's knowledge, there have been no proposals along these lines.

A GRAPHICS LANGUAGE BASED ON SEVERAL COMMERCIAL SYSTEMS:

One popular process control vendor system uses a command language to display process graphics, both data trends and pictorial displays.

What are the elements of a process display? There are process elements, such as tanks, valves, pumps, etc. These are connected by pipes or lines and lines with arrows to show direction. Most process elements have a value associated with them. The displayed values should be updated frequently.

At least one "language" is a series of commands and position indicators which are converted to the BASIC language and then interpreted by a normal BASIC system. BASIC is also used to do the controlling and sequencing of the plant elements in some systems.

IBM presented a paper on a "Picture Building System" in 1980 at the SHARE meeting. It seemed as if many of the concepts in other vendor systems were presented at that time. I do not know the status of the language within IBM, but I have not heard any further discussion on the part of IBM.

My proposed commands are similar to some vendor systems and the IBM system (as I remember them). The commands are listed and then the details of each command are presented. An example of how they would be used to construct a process display in use at a typical plant is presented.

COMMAND STRUCTURE:

The commands have a uniformity in arguments as far as possible. For example, every command used in a display starts in the same way:

```

line label (a number, optional)
command
color
lower left x-position
lower left y-position

```

Some commands, such as WINDOW or PLOT, cannot have a uniform structure since those commands are not display element commands, but could be thought of as "format" commands.

COMMAND SUMMARY:

The commands in some vendor or IBM system are similar to the commands in this system. The commands for this language are:

```

ALINE color,x1,y1,x2,y2,t,m
ARC color,x1,y1,x2,y2,r,t,m
BACK color,x1,y1,x2,y2
CALCULATE
CIRCLE color,x1,y1,r,t,m
CONNECT color,label1,label2,t,m
CONT x3,y3,x4,y4,x5,y5,x6,y6,x7,y7,x8,y8
DEBUG on/off
DO n COMMANDS, i TIMES,DX=dx,DY=dy
FREQ seconds
INCLUDE name,x1,y1,x2,y2
INTERSECT color,label1,label2
LINE color,x1,y1,x2,y2,t,m
NAME display name, (description)
PLOT STATUS
PLOT TREND
PLOT DATA
PLOT DEV
PUMP color,x1,y1,s,391
ROTATE a
TANK color,x1,y1,x2,y2,t
TEXT color,x1,y1,s,this is the displayed text
TRACE on/off
VALUE color,x1,y1,s,249,w,d

```

VALVE color,x1,y1,s,27

WINDOW name,x1,y1,x2,y2

where (x1,y1) is the lower left hand coordinate of a line, the first letter of text, a value, or some process symbol such as a tank. This is typical graphics representation, even back in 1964 on the old Calcomp incremental plotters.

In all instances, "x" represents a position on the horizontal axis and "y" represents a position on the vertical axis.

The position (x2,y2) would be the end point for a line, or the upper right corner of a rectangular area.

Any argument which begins with the letter "L" is a line label, denoted as "labell", etc. in the above summary. Line labels begin in column 1 of the command and are merely numbers. We can differentiate a line label from a command since all commands start with a letter, and the line label is a number.

The (description) option in the NAME command is for the menu display.

The parameter "a" in the ROTATE command specifies the angle, in degrees, through which the subsequent display elements are rotated (in a clock-wise direction from the horizontal).

There is a variation of the (x,y) pairs for coordinates which is based on some computer language conventions. If one wants to continue some function, such as drawing a line or displaying text, it is redundant to have to specify the (x,y) coordinate pair for each piece of the figure. In many instances, the piece of figure is just a continuation from where the last piece of figure was displayed.

In some computer languages, the use of the character * is used to indicate that you are to "continue from where the you were last". This would be useful in graphic displays so that one could set up an entire line and be able to change the line position by only changing one beginning coordinate.

The * notation also allows arithmetic to be done on the * value, for example, "* + 20" means a position 20 units from where the previous instruction ended. Thus, one can build a series of rather complex pieces and have the whole figure based on one or two coordinates.

This gives a great flexibility to the display designer and allows him to move large sections of the display with a minimum of changes of the (x,y) coordinates.

The system date and time are useful features in any display. Special commands such as SYSDATE and SYSTIME would display the

date and time for the values displayed. The time would be updated, along with any values displayed, at the same time interval as value updates. (It may be necessary to place the date, time and display ID in some fixed position on the screen. For purposes of generality, the ability to place this system data on a screen should exist.)

DATA PLOT OPTION SUMMARY:

There are several options which are valid only when a plot is requested. These are:

INTEGRATE option

PROJECT option

The INTEGRATE has an effect only on one dependent variable (y-value).

The PROJECT option is used only with a time axis. An extrapolation into the future can be specified with this option on the TIME command in a PLOT.