A design exercise performed by human factors specialists is described. In this exercise a front-of-screen simulation of the Interactive Chart Utility was written before a working prototype was available in order to draft and test a series of on-line instructional (HELP) panels for incorporation into the final product. Trials were run in which the keyboard activity and utterances of naive subjects were recorded for later action replay, before and after redrafting the simulation. Three objective measures to detect the resulting improvement are considered, and the most robust identified.

Software simulation as a tool for usable product design

by I. A. Clark

A software developer is usually extensively equipped with tools for testing the programs he or she writes and is supported by service groups, such as Product Assurance in IBM, that provide further testing. A program can be tested to see if it actually runs as soon as it has been written. It is unthinkable that a software product would be shipped without first ensuring that it actually ran on the machines for which it was intended.

A job aid such as a manual or an on-line assistance facility (called a *HELP facility*) is the counterpart to the program for the human user. Yet it is no secret that such job aids have been written and shipped with nothing like the testing enjoyed by the program code.

Programs are written in languages that follow rigid rules to specify instructions to a computer. For each computer there is also a written functional specification available to the programmer, who is already intimately familiar with its principles of operation. There are relatively few variations on the basic model, which further simplifies the task of writing an effective program for a given machine.

The same cannot be said for the human user. The developers of a product hope that it will appeal to a wide audience, the wider the

better, and the manuals for the product are translated into many different languages. But even among those who speak English, the same words can have quite different meanings to any pair of people chosen at random, or even to the same person in different contexts. For example, to an electronics engineer the meaning of the term "bus driver" will be quite different depending on whether he encounters it at his workbench or in the street. We might then pose the question: Is there any hope that the text of a written manual or an on-line HELP facility can be verified to perform as it is intended, while there is still time to rectify defects?

Today provision for on-line HELP facilities in an interactive product is often made during product development. Printed manuals can be written to a large extent independently of the program, but not so an on-line HELP facility. It must be developed in close conjunction with the program code. The style of text suitable for a manual turns out not to be appropriate for a HELP facility. Thus, when the developers of the IBM Graphical Data Display Manager (GDDM) and Presentation Graphics Feature (PGF)[1] decided to incorporate a HELP facility in the Interactive Chart Utility (ICU), a part of PGF, they invited the Hursley Human Factors Laboratory to draft the text of the HELP facility.

The ICU is intended to permit nonprogrammers to construct business charts in color, or to view and alter existing charts which might have been generated by a program or by somebody else. The repertory of business charts includes line graphs, surface charts, histograms, bar charts, pie charts, and Venn diagrams. The resulting chart is displayed on an IBM 3279 color display device.

The ICU user creates or modifies a chart by causing so-called "menus" (really overtypable forms) to appear on the screen and then changing the values of fields on the forms by typing new values over them. Fields are never blank, unless blank is a permissible variant, so that even a fresh set of forms will appear with initial values (defaults) in all fields. Once the user has typed in suitable sets of coordinates representing his or her data, pressing a given key (the "DRAW" key) will usually produce an adequate chart without any of the many fields under the user's control needing alteration.

The ICU was envisaged for use by scientific and nonscientific professionals, managers, and their secretaries. No special training was planned to be given, so that completely unknowledgeable personnel would use this tool with no more help than that afforded by the on-line HELP facility. There were no plans at the time to produce the printed self-tutor that was subsequently provided.[2]

Menu panels were arranged in a tree structure, which determined what panels were accessible from a given panel. HELP panels had to be "daughters" of the menu to which they referred, so that all notion of a unified manual with a structure of its own had to be dismissed. Originally there was to be one HELP panel per menu panel, but this constraint on the HELP drafter was soon relaxed, and a series of continuation pages was permitted, through which the user could only proceed forward, not being able to backtrack or browse.

The design was already far advanced, thus limiting what could be done within the available time. For instance, the menu wording might be altered in conjunction with drafting the HELP panels, but the degree to which the flow between panels could be altered was strictly limited. The Human Factors Laboratory set out to furnish the best solution that could be achieved within the constraints imposed by the system design. The time limit was alleviated somewhat because a simple simulation of the ICU was already in existence, previously written by the Human Factors Laboratory for its own purposes.

Method used to perform exercise

Twenty subjects obtained from an employment agency used a simulation of the Presentation Graphics Feature, Interactive Chart Utility (ICU) to perform the task of altering a color business chart in a prescribed manner. Subjects worked in pairs, one pair per session.

In consultation with the developers, a series of simulations representing redesigned versions of the ICU and its HELP facility were built, starting from the published external specifications of the ICU. The simulation apparatus consisted of the Virtual Machine/Conversational Monitor System (VM/CMS) EXEC interpreter, invoking the IBM Input/Output Display Facility (also known as IOS3270).[3]

The original design made use of certain terms when furnishing its options to the user (e.g., X axis, Y axis). Others were under consideration as possibly more accurate terms to use instead (e.g., abscissa, ordinate). The designers and experimenter had their suspicions about whether users would understand the terms or not. Therefore, before they used the ICU, subjects were given a word-comprehension test.

In the test (Figure 1), 17 suspect terms were singled out and listed in proximity to an assortment of drawings (Items 1 to 9). Each term was exemplified at least once somewhere among the items shown. However, to reduce the possibility that an unknown term

Figure 1 Questionnaire used for word comprehension (The item-numbers 1–9 are only for reference and did not appear in the copies given to the subjects. Four items are identified correctly as the subjects were invited to do it.)

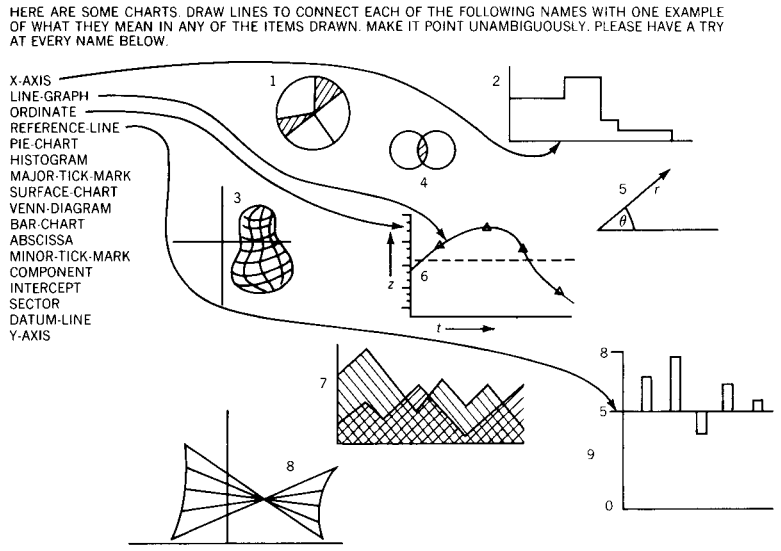


Table 1 Number of subjects (out of 20) who identified each term correctly

Term	*Subjects*	*Percentage*
Pie chart	16	80
Bar chart	15	75
Line graph	12	60
Y axis	12	60
Venn diagram	11	55
Histogram	10	50
X axis	9	45
Sector	8	40
Ordinate	6	30
Surface chart	4	20
Major tick mark	3	15
Datum line	3	15
Reference line	2	10
Intercept	2	10
Minor tick mark	1	5
Component	1	5
Abscissa	0	0

could be hit upon by elimination, a few false trails were laid, notably Items 3, 5, and 8. Subjects were not told how well they had done until the end of the experiment, so that the effect of lack of knowledge of these terms could be observed in practice.

Correct guesses for the terms are presented in Table 1, which shows for each term the number of subjects (out of 20) who got it right. Terms are ranked in the order of correct guesses.

Table 2 Type and occurrence of recurrent errors in 10 trials with a chart utility simulation[1]

Errors	*Trial*										*"t"*[3]	*signif?*
	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*		
1. Confused PF*n* with typed "*n*"	4	4	2	1	2	—[4]	2	—	—	—	1.84	
2. Accidental termination	1	—	—	1	1	1	—	—	—	—	2.53	YES
3. PF3/return to parent panels	1	2	1	4	1	4	1	—	—	—	2.46	YES
4. HELP access/return	—	1	—	—	—	—	—	1	—	—	−0.03	
5. Misplaced response in HELP	1	2	2	2	—	1	—	—	—	—	3.20	YES
6. Incomplete simulation/bug	2	1	7	2	2	5	3	2	1	2	0.95	
7. X/Y	—	—	—	1	—	—	—	—	—	—	0.80	
8. Stuck in DRAW	—	1	—	—	—	2	—	—	—	1	0.53	
9. Stuck in home panel	1	1	—	1	1	—	—	—	—	1	1.26	
10. ENTER misplaced/redundant	2	2	3	4	2	1	1	1	5	—	0.57	
11. Typed into wrong field	3	2	1	—	—	5	2	—	—	2	0.76	
12. Typamatic key	2	1	—	—	—	—	—	—	1	—	0.53	
13. Misread key tops	2	—	—	—	—	—	—	—	1	—	0.18	
14. Misunderstood panel	2	1	3	1	3	3	1	4	1	2	0.22	
15. New menu construct	—	1	—	1	—	—	—	—	—	—	1.26	
16. Misunderstood word meaning	5	1	3	2	2	—	—	—	—	1	2.13	YES
17. HELP confused/misled	1	1	5	1	—	1	—	—	3	2	0.23	
18. HELP actually helped (not err)[2]	—	—	—	—	—	—	1	1	—	2	3.09	YES

Notes:

1. Trials A to F were conducted with a simulation of the original proposal, called PPI04. Trial G used PPI05, H and I used PPI05A, and J used PPI06, which were progressive refinements of PPI04.
2. Item 18 is not an error, but was a recognizable event, and usefully analyzed the same way.
3. The last column is the Student "t" statistic, with eight degrees of freedom, used to reject the hypothesis H_0 that the progressive "improvement" of the simulation over the last four trials, G to J, had no significant effect. H_0 can be rejected with 95 percent confidence if $t > 1.86$. Where this is so, "significant?" = YES. Elsewhere one should conclude that either the measure was not discriminating enough to reveal the improvement (e.g., 1, 9) or there was no improvement (e.g., 14, 17).
4. Dash is used in place of zero errors to aid readability.

One subject was then assigned to operate the keyboard and was told to play the role of a general clerical worker faced with the task of using an unfamiliar system that drew business charts. This role was, of course, a fair description of the situation she found herself in. She was told that the system displayed instructions ("HELP") whenever a certain key was pressed (namely, a key marked PF1, which was pointed out to her). Beyond that she knew nothing about how to operate it. That being so, she was to imagine she had fetched a friend to help her (the other subject).

A hand-drawn example of a business chart was exhibited. This chart was marked with a number of corrections in blue to show how it was to be altered. Next, a fair copy of the end result was exhibited. Both were kept in view of the subjects throughout the experiment. The session-recording apparatus was then set in motion, and the trial was concluded when the one-hour recording tape ran out. On another questionnaire subjects were asked separately to rate subjectively the ease or difficulty of doing various aspects of the task.

A time-stamped computer listing was produced by the simulation apparatus, recording the state of the simulation at each point when ENTER or a program function (PF) key was pressed. Subjects' voices were recorded on one track of a two-track (stereo) tape recorder. On the other track all keystrokes were recorded by a data-logging device that intercepted the lead connecting the keyboard to the display head. The keystroke record so obtained was complete enough to allow accurate real-time replay, in synchrony, of both voices and screen activity. The latter was possible because the sequence of recorded keystrokes could drive the host computer through precisely the same sequence of states as during the actual session.

The experimenter afterwards replayed parts of the session to the subjects and asked them to recall what they had in mind when they did or said certain things. Later the experimenter replayed the entire session to himself, annotating a printed version of the log produced by the simulation, which he then used to redraft the HELP text, or to propose design changes to be incorporated in subsequent versions of the simulation. It was found important to do this the same day as the session itself.

Later an independent reviewer repeated the process of listening to the tapes and annotating a printed version of the log. The error counts of Table 2 are taken from his records, rather than the experimenter's.

Scientific background to the exercise

The problem of whether concurrent verbalization affects the way a subject goes about the task has been treated elsewhere.[4,5] The trick of having two subjects—the less assertive operating the terminal, the more assertive helping—seems to ensure that subjects actually verbalize, and in a more natural fashion than can be achieved by continual questioning of a solitary subject, as was done by Hammond *et al.*[4]

The experimenter found himself reinterpreting much of his memory of the session in the light of the resulting faithful action

replay. He could put himself in the subjects' position and get a much clearer understanding of why they did and said what they did, than he was able to at the time. He was surprised by what he had missed. Even if little was said, inarticulate expressions of dismay, satisfaction, and frustration carried important information, which could be interpreted by reconstituting the subject's environment around oneself.

This method is probably as near as anyone is able to get towards seeing the task through the subject's eyes. It also seems to achieve directly the goal of much human factors activity which at present communicates its results indirectly via technical reports, namely to supply information to the process of designing man-computer interfaces.

Unfortunately, it is difficult to quantify or evaluate this sort of activity, except, of course, subjectively. It is important, therefore, to draw objective evidence out of the session records to support the strong impression gained by the experimenter/drafter that he was truly perceiving the causes of difficulty and actually doing something to improve the design.

field testing

Johnson and Baker[6] accurately describe the situation confronting the human factors engineer engaged in what they call "field testing." It is not, as they say, a simple extension of the laboratory into an operational setting. The behavioral psychologist who ventures into this area finds the familiar forensic weapons of meticulous experimental control missing: a well-defined population properly sampled, an exhaustive list of variables to be controlled, and the liberty to replace the task under investigation, which everybody recognizes, with an abstract paradigm lending itself to better control of certain variables, even if its connection with the real-life task may be obscured.

As it happens, this experimental approach (sometimes known as reductionism) serves chiefly to increase the distance between the human factors engineer and the designer he seeks to inform. Chapanis[7] goes so far as to call in question the relevance of much laboratory work, performed in the best reductionist tradition, to any real-life situation at all.

However, much in the history of behavioral science points to the need for ever-stricter experimental control in the search for scientific truth to ensure reproducibility of experiments and for reliance upon objective measures wherever possible to reduce observer bias.[8] If any further prompting were needed, such exposures as that of the recent one of Sir Cyril Burt,[9] at one time the doyen of British psychology, have tended to harden the attitudes of academic behavioralists against subjectivism in all its forms.

Nevertheless, the human factors engineer who agrees to work with a deadline, subject to constraints not imposed for a full scientific study, does not need to jettison all philosophical foundation for his or her work. Nor has he or she nothing to learn from the laboratory methods of behavioral psychology.

Without belittling the work, it is properly described as nonscientific. Design, in particular the design of a HELP facility, is a creative activity by one person for the benefit of another person. No reference need be made to the concept of scientific truth to justify such activity. The designer uses his perception, aided by tools, to judge how to make appropriate design decisions. Science begins and ends with the objective evaluation of those tools, and a study of the data they yield.

There is, of course, a science of decision making. This science evaluates the strengths and weaknesses of proposed tools for making decisions, such as a statistical test. The designer may or may not use this tool in coming to a decision. Ultimately the only criterion for judging whether the correct choice was made is the success or failure of the design. The only philosophical motive for using a scientifically approved tool is to furnish some assurance, before completion, that the design will be the better for doing so.

It is important to clarify this matter when judging a design exercise of the sort being described. The present exercise leaned heavily upon the laboratory techniques of the behavioral psychologist. It was run with apparatus and methods similar to those of the reductionist experiments described by Hammond *et al.*[10] Yet it is not to be compared with these experiments in a scientific sense.

To begin with, there is nothing like the same level of control of variables. In the case of Reference 10, this took months of pilot studies and redesign of the task, as progressively more experimental variables were brought under control and conditions multiplied. With a balanced statistical design used, each new variable to be analyzed doubles, and maybe triples, the number of subjects that must be run. With six (two-valued) variables controlled in this way, 64 subjects are required (none to be reused), each subject generating data in the form of an hour's terminal activity plus before and after questionnaires.

The worst damage to realism, however, comes from controlling all the other variables that are not going to play any part in the analysis of variance and must therefore be kept constant, such as what the experimenter says in the course of the experiment, how he answers requests for clarification, the response time of the computer, the possible different ways of completing the task, etc. It can safely be said that there is no future for this sort of exercise in the time scale of a typical development project.

By contrast, the present exercise employed just two groups of trials. The first control group of five trials used the same version of the simulation. After that, each successive trial was run on a new version, redrafted to address the difficulties subjects were encountering.

Statistical analysis of such an experiment is strictly limited in scope, but it is possible to detect, using Student's t-test, whether the redesigned exercise had any significant effect on the subjects' performance. The word "detect" is used, because the t-test is being employed here not to demonstrate the scientific truth of any proposition, but to detect a signal in noise, the signal being the beneficial effect, if there is any, of redesigning the simulation.

For this purpose a small sample size actually lends credence to the significance of our result, if that result is to reject the null hypothesis (i.e., the hypothesis that the redesign had no effect). The effect of a larger sample size is simply to make it more likely that a weak signal is detected, i.e., that a weak (albeit genuine) impact of the redesigned exercise on subjects' behavior shows up as statistically significant.[11] With a sample size of just ten trials, only the stronger effects make themselves apparent (statistically significant) above the "noise" of random variation.

A good analogy for this use (or abuse) of behavioral techniques arises from contrasting the use of, say, a proton magnetometer in a physical standards laboratory to measure a fundamental property of matter with its use in a quarry by police to detect buried metal objects (as the author once encountered). In the latter case, the conditions under which the instrument was used were such as to invalidate any scientific generalization from the result. Yet it did discover a buried object.

significance of paper

The scientific, as opposed to the methodological, significance of this paper thus lies in its answers to the following questions:

- What signals, if any, were objectively detected during the exercise?
- What is the likely source of random noise that could conceal signals?
- What systematic noise might there have been to produce false signals?
- What subjective insights were gained concerning user difficulties? (We may ignore those for which there is no objective supporting data.)

Analysis of results

As stated previously, an independent reviewer, who had not been involved in the actual exercise, was employed to rate the sub-

jects' performance. He did this by observing the action replays of the experiments, simultaneously annotating a printed version of the simulation session log with instances of errors. A full voice transcription as by Hammond *et al.*[4] was not produced, although this is recommended if the time and resources are at hand. Otherwise, error events can be missed or misinterpreted. Table 2 shows numbers of errors counted from this reviewer's annotations.

The reviewer also marked a task-breakdown sheet for each pair of subjects to judge which subtasks were completed. The experimental task was broken down into 12 normative subtasks, each of which broke down into an average of six sub-subtasks.

For the last four pairs of subjects, the simulation went through the following redraftings:

- PPI05 (introduced the "Home" program function key to simplify navigation, some panel rewording)
- PPI05A (a complete new draft of HELP)
- PPI06 (redesigned "Exit," reworded navigational parts of panels)
- PPI06A (as PPI06 but with a brief terminal tutorial inside HELP)

Objective results

The number of subtasks completed varied widely and showed no significant difference between the first six and the last four trials. Neither could any significant effect be discerned in the subjects' subjective rating of what they found to be difficult. Opinions varied widely even between members of a pair. The HELP facility itself was considered easier on average to use than any other item on the questionnaire, but here the wording defeated any useful interpretation of the result. The HELP facility was found to be "easy to use," not necessarily "helpful," as the questionnaire should have asked.

It seems, therefore, that signals from two of the most popular measures of improvement to the design were submerged in the "noise" caused by uncontrolled variables. In the experimenter's opinion, most of the noise arises from the wide variation in the subjects' ability and from the different things that were said and done in each session. The first variable is hard to control, the second relatively easy but only at the expense of artificiality. Typical solutions to the latter would entail designing the task carefully so that there was only one solution path, or else driving the subjects along a chosen path by forbidding deviations.

However, when we examine the recurring errors made by subjects according to their type and occurrence, we do find appreciable signals (Table 2).

In hindsight, both of these results are what might have been expected. If we consider Sackman's observations[12] of ratios in the order of 30:1 between the productivity of the most and least able of the normal population when using time-sharing systems, it is absurd to expect that a mere ten trials will yield statistically significant task-completion effects when there has been no control of the subjects' ability, either by selection or by measurement.

Nonetheless, the experimenter/designer was in a superb position to observe recurrent types of error and to redesign the simulation specifically to attack them. Not surprisingly, the greatest improvements came in those areas where greatest effort was devoted, e.g., in navigating between the different panels. This subtask proved to be unexpectedly difficult with the first version of the simulation.

No significant effect arises in those areas where there was little hope of doing much good, such as redundant ENTER keystrokes. These areas were considered to cause little damage to the ideal task structure.

It is important to ask first how much a study of errors and word comprehension contributes to designing systems that can be used productively. A product planner, and perhaps the purchaser of a product, will be interested in business cases based on percentage productivity gains due to a given line item. However, as we have said, it is difficult to measure productivity gains directly and even more difficult to establish statistically their true cause. Frequencies of certain sorts of error are a much more sensitive measure. They contribute to productivity in an obvious way, even if the relationship is complex. Moreover, there is some hope for a mechanism to explain how and when they arise.[13]

Nevertheless, in the case of the ICU, it is important to remember that productivity was a secondary issue. Acceptability is much more important. The ICU is aimed at people who have some discretion as to whether they use a computer in a particular way or not, and want to use one in a way they have not done before, namely to draw colored charts of business data. The question is, can they do so at all? Never mind whether they do so efficiently, at least to start with. Some sacrifice of efficiency may be permissible in order to assist them. The greatest barriers to their uptake of a novel system come in the first hour of use. If they find that they cannot surmount this hurdle because of obscure terminology or frustrating errors, they are unlikely to persevere with

using the ICU. That is why we concentrated on naive subjects and their reactions during initial exposure to the product.

It also serves to warn that, whereas our choice of subjects, consisting as it did of people with no particular skills who were obtained from an employment agency, may have been appropriate for an investigation of the ICU, it may not be appropriate for a system designed for a specialized class of user. In such a case the employment agency would have needed more precise instructions on the type of staff to supply, an acceptance procedure would have been advisable, and a course of preliminary training probably necessary.

Underlying causes of error

In the absence of further (scientific) experiments, the underlying causes of error have to be a matter of conjecture, although we may ignore those hunches that have no counterpart in Table 2.

The preliminary word-comprehension test showed that the subjects had a poor level of knowledge of special terms related to graphs and charts. Nearly all subjects could recognize a bar chart and a pie chart. Some thought that a histogram was just another name for a bar chart. Nobody knew what "tick marks" were (a draftsman's term for graduation marks on an axis). The terms "X axis" and "Y axis" were fairly familiar, whereas more precise terms for the same things, "abscissa" and "ordinate," were not. Subjects groused freely about incomprehensible jargon.

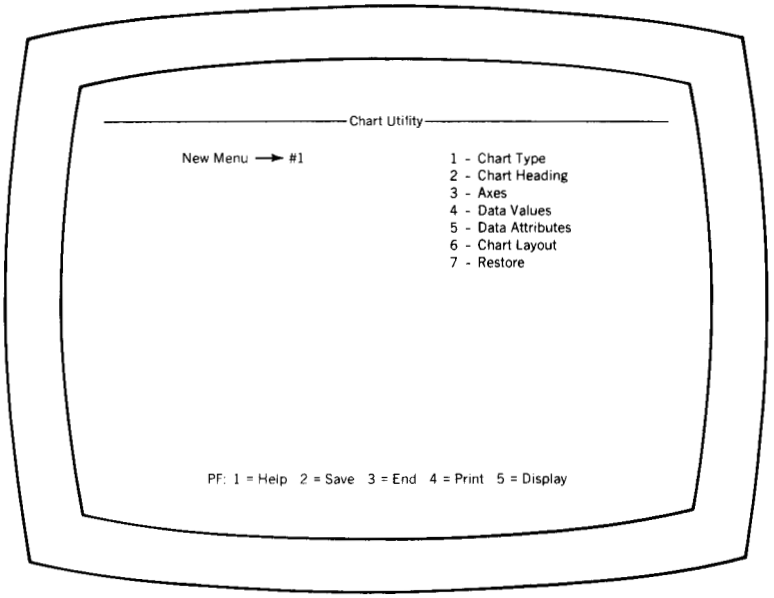
Surprisingly this lack of knowledge did not appear to contribute to the difficulty of using the ICU. Wherever an unfamiliar term such as tick mark was used, it was easy to clarify what this meant in the associated "HELP" panel, in this case by a simple diagram. Subjects appreciated diagrammatic explanations where these were feasible. This is not necessarily a recommendation to use them, however. There seems to be a greater possibility for a user to misunderstand a diagram than the words in a text, although this does not detract from the greater appeal of a diagram to the user.

In fact, few errors arose primarily from the difficulty of the task or unfamiliarity with drawing graphs. Rather they stemmed from unfamiliarity with the display device itself (especially when the keyboard locked) and from the task of what we shall call interpanel navigation.

There were two problems associated with interpanel navigation:

- Calling up the appropriate panel containing the field to be altered to achieve a given effect

Figure 2 The home panel, as originally specified, from the version of the simulation of the ICU called PPI04



The # sign signifies the start of an overwritable field.

- Returning to a state to undertake the next subtask

The latter problem seemed to be the more time-consuming and error-prone.

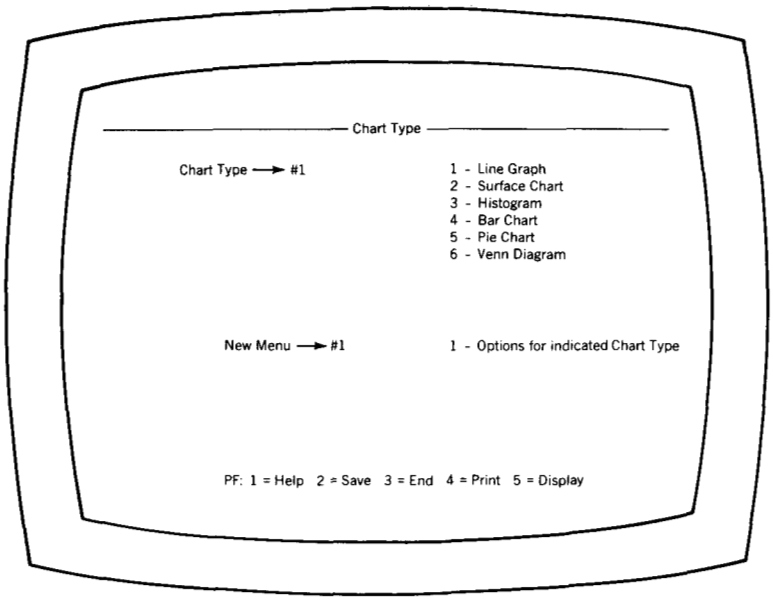
ways to produce new images

Subjects can be forgiven for expecting just one way of making a new image appear on the screen, or at least a small number. However, several different ways are available in the ICU utility. Not surprisingly, subjects were often unsure which to use. The different ways are presented below in the order in which they were first encountered in a typical session. This order is important, because naive subjects are inclined to induce "rules" for what to do next from what has been successful in the past. Once induced, such rules are hard to displace.

Press a program function (PF) key. For example, in pressing keys to go to another panel from the "Draw" panel, PF03 is for "Home" or PF01 for "Help." Other PF keys will later yield unique panels, e.g., PF02 for "Save," PF04 for "Print," PF05 for "Draw," PF12 for "Home." The subject quickly induces the rule that these PF keys (usually) work from any panel.

Press ENTER. The naive user soon calls for on-line assistance. In "Help" (but nowhere else) pressing ENTER shows the next page of notes.

Figure 3 Chart-type panel, as originally specified, from the version of the simulation of the ICU called PPI04

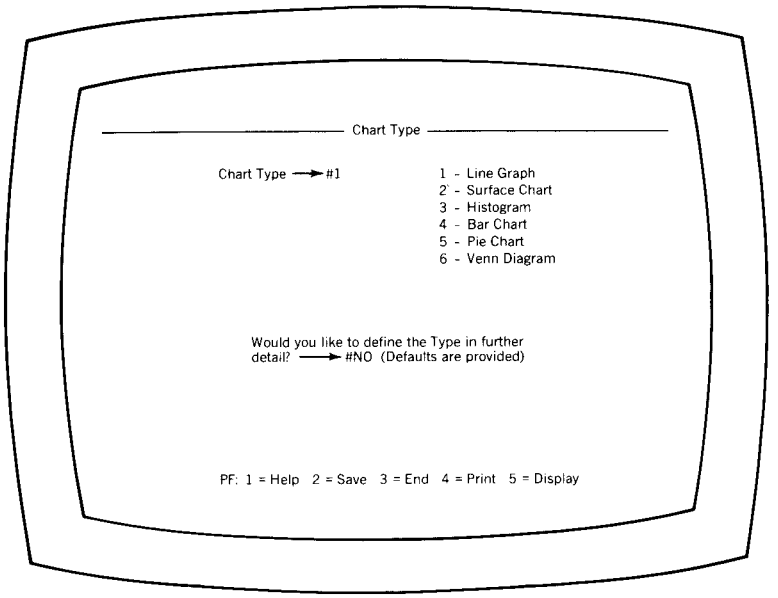


The # sign signifies the start of an overwritable field.

Type a number into the first field on the screen, then press ENTER. By now the subject is being shown the Home Panel (Figure 2). The behavior of this field occurs nowhere else since it is a single-character autoskip field and the only one on the screen. Thus, the cursor stays put after each character keystroke. The subjects' experience up to now may prompt them to do a number of things, such as type a number (e.g., 3) and immediately afterwards press PF03, or press PF03 only (say) to make the menu choice numbered 3.

Type YES *in preference to* NO *in answer to a question*. For example: Do you want to do such-and-such? This construct was introduced in the PPI05 redraft and those following in place of the original PPI04 construct exemplified in Figure 3. The New Menu Field shown in the figure serves to route users to other panels further down in the hierarchy. All subjects failed to comprehend the meaning or intended use of the New Menu Field, but it is actually a collapsed form of Figure 2, i.e., with only one menu choice. In the strictest sense, it was a double, not a single choice, namely either type a 1 or leave it blank. Introducing yet another way of raising a new panel was the lesser of two evils. Subjects implicitly knew what was expected of them here (see Figure 4). This panel is similar to the corresponding panel in the simulation PPI06 of the ICU. It is used consistently wherever there is the choice of only a single daughter menu to see.

Figure 4 Chart-type panel, similar to the version in the finished product, but formatted for comparison with Figure 3



The # sign signifies the start of an overwritable field.

Type e over a leading numeral, e.g., to turn 001 into e01. This instruction was needed in the Data Components panel in order to reveal more details of any given component. Subjects found this a somewhat curious thing to do, and made a variety of errors doing it. Similar operations are needed to add to, delete from, and reconfigure the list of components (and other lists).

Scroll the panel. This operation is done by pressing (as appropriate) PF07 (Back), PF08 (Forward), PF10 (Left), or PF11 (Right). It was not obvious to subjects that the result was to show more of the same panel, as opposed to a different panel, so these operations are included here as if they were part of interpanel navigation.

It may be that all these different ways of going from one display to another are unnecessary. In hindsight it may be apparent how to replace them all with a single construct which would rapidly become familiar to the user and would invite comparison with (paper) official forms, etc. However, the experiment did not take into account such hindsight, and only superficial modifications to the interpanel navigation apparatus were undertaken. But even these significantly improved subjects' navigation, as shown by the reduction in Type 3 errors (return to parent panel) in Table 2.

The greatest benefit seemed to come from introducing the concept of the "home panel" (the topmost menu in the hierarchy) and creating a new function (pressing key PF12) to get the system back to the home panel from any state, even from within the online HELP facility. Subjects rapidly took to using this function and seemed to derive reassurance from it. So long as it really did work for any state of the system, subjects gained confidence that they would not get stranded or lose their way within the panel hierarchy. This confidence was noticeably lacking during earlier trials.

The problems of drafting HELP panels

Since the main purpose of the exercise was to draft a set of HELP panels, it is to be expected that the first attempts were inadequate. The chief pitfalls encountered were as follows:

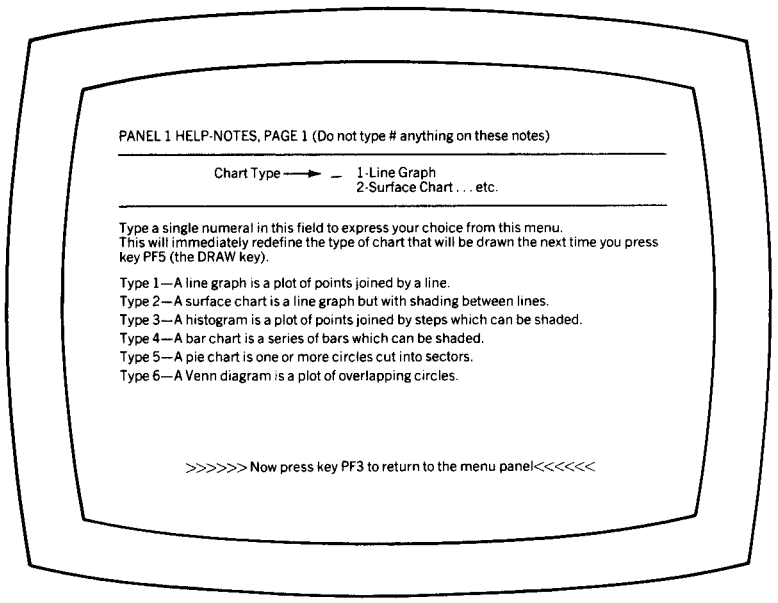
Lack of a point-to-point relationship between HELP panel and menu panel

A serious problem arises with HELP panels when users are trying to describe the contents of a separate screen display. If they can, naive users will tend to have both panels visible simultaneously so that they can glance rapidly from one to the other. Some will go so far as to place their fingers upon the material (e.g., screen and printed manual) to relate them point-to-point.[14] Where they do not actually use their fingers, they are likely to be employing some functionally equivalent mental imagery.

Whatever the mechanism, robbing subjects of this means of operating by ensuring that the HELP material has to be viewed through the same screen as what it describes causes them acute difficulty. Subjects will call up a HELP panel, then puzzle over it, seemingly having forgotten why they wanted to look at it in the first place. Or else they will return to the menu panel, and then appear to forget what it was that they had discovered from HELP. They will scan a HELP panel with a definite problem in mind (as their taped utterances confirm) yet seem unable to recognize the relevant explanatory paragraph.

What seemed to be lacking was a positional landmark in the HELP text that could be recognized by shape (rather than by meaningful content) as relating to the given field on the menu that subjects are inquiring about. Many ways of achieving such a landmark are available, but the one chosen was preferred because it made no new demands on the program coding. It consisted of heading each page clearly with a reproduction of what the menu field looked like, so that the reader could pass rapidly over irrelevant pages.

Figure 5 HELP panel, page 1, from the version of the simulation as redrafted at the end of the exercise, which replaced Figure 6



The # sign signifies the start of an overwritable field, namely word "anything" at the top of the screen. This was provided to prevent the keyboard from locking in case the user erroneously typed characters.

See Figure 5 for an example. There was one such page per field on the given menu. Subjects were observed (and recorded) to use this method exactly as intended.

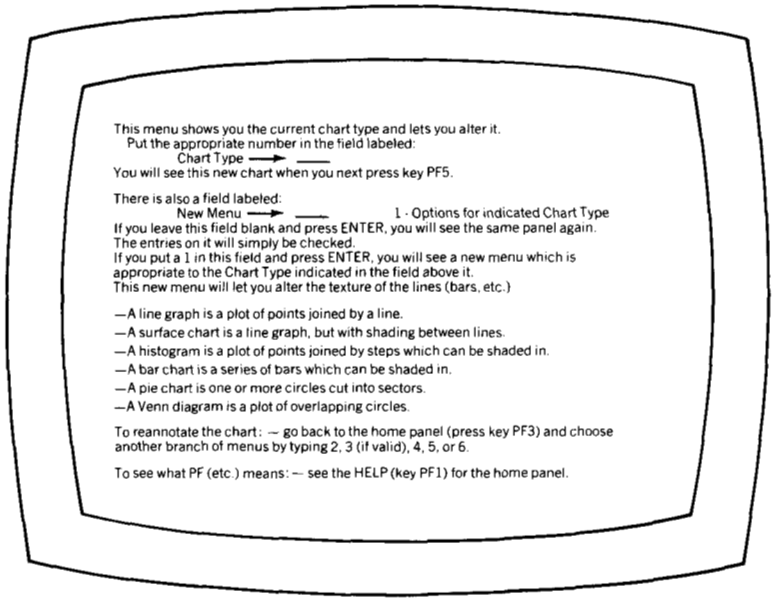
However, there were undesirable side effects. Some subjects failed to realize that the picture of the field was not the field itself and tried to type over it. It was suggested at the time that the HELP panels should themselves serve as alternative data-input panels, but the program designers could not accommodate this request.

Confronting the user with a solid block of text

Figure 6 exemplifies the first draft of the HELP panels, which presented a block of text to the user by trying to fit all the available information on one page. Subjects found it neither helpful nor reassuring. Somehow the user must be able to avoid having to read through a mass of material to find what he or she wants. This item is complementary to the previous topic. Whereas the problem there was to allow the user to home-in quickly on what was wanted, before short-term memory decayed (1 to 2 seconds), here the problem is to recognize the beginnings and ends of blocks of text so that the user's eye can scan quickly over irrelevant material.

To achieve this, HELP panels were stylized as far as possible; thus, after seeing one or two pages, the subject could recognize at

Figure 6 HELP panel for the chart-type menu shown in Figure 3



a glance the significance of the format of each grouping of words, or typographical construct. It seems that shape and visual pattern play a more important part here than the actual meaning of the words. Justification of text, although it looks neat, seems not to help legibility. It is better if each sentence can start a new line.

Pitching the reading age of the English text too high

HELP panels seem to need couching in a style suitable for a six- or seven-year-old, even to having one sentence per line. Adult users can, of course, understand greater prolixity, but not, it appears, keep hold of the problem occupying their minds at the time. The "fog index"[15] was used as a rough and ready yardstick.

Telling the user to do something, but not there and then

Consider the two forms of words in this instruction:

- Pressing key PF5 will cause the picture to be drawn.
- To draw the picture, press key PF5.

The first is couched in passives, participles, and noun phrases as if it were a vague remark. The second is an order. Subjects said, in so many words, that they preferred the latter since it stood out for them as an unmistakable signal against the "noise" of all the new material that was bombarding them.

Unfortunately, if they were told to do something, such as press a certain key, subjects were apt to do it there and then, whereas the designer intended them to return to the menu panel first. This behavior proved so hard to modify that it was felt that it would be better to concede to users' obvious expectations and make the PF keys behave as they would if the menu panel were showing.

Users were also in the habit of typing characters while looking at the HELP panels. Originally these had no overwritable portion, which resulted in the locking of the keyboard when users took such action. On this type of display device, typing into a so-called "protected field" causes the keyboard to "lock." An indicator light comes on (which the user may not notice), and the screen becomes unresponsive to whatever else the user does until he presses a key marked RESET.

This feature of the device caused users so much annoyance that it proved advisable to avoid the likelihood of the keyboard locking as a result of normal user behavior. A dummy field was provided on each HELP panel as part of a conspicuous message saying: "Do not type anything on this panel." This addition seemed to overcome the difficulty. However, it was invariably the helper, not the keyboard operator, who quickly recognized when the latter was doing the wrong thing.

In case the reader wishes to study the HELP panels of the finished product as an example of putting these rules into practice, it should be mentioned that they are not entirely as described here. Alterations were needed to suit already written program modules, and late changes were made to the design.

Incidentally, our study indicates that the word "help" itself appears to be a poor choice of name for an assistance or instructional facility. Help seems to carry connotations of distress or for use in emergencies only. Such a connotation may underlie the hesitancy of subjects to use it; instead they prefer to sit for minutes in puzzlement. The matter needs further investigation.

Requirements for an adequate simulation tool

The case for writing a front-of-screen simulation of a proposed interactive product has been made by Clark[16] and by Meijer.[17] Meijer described the use of a tool originally developed for writing computer-based training courses. The philosophy is to track the design from the earliest stage to serve as a discussion medium between designers and reviewers, including human factors engineers. Eventually it can be used as the basis for training courses for operators of the product.

Table 3 Main points to consider when choosing a simulation tool

Low probability of introducing bugs as result of a modification to the simulation
Quick to alter a large number of similar panels
Quick to see the end image of a panel when altering it
Ability to replay a session in real time
Ability to edit such a replayable session, including the voice/keystroke tape, in order to compose a sequence of highlights for presentation to designers
Ability to run an incomplete simulation and add to it while it is running
Ability to manage groups of related panels
Ability to annotate screen displays produced as record of session
Ability to print a fair copy of a panel with highlighting for a report
Ability to load a panel layout from other systems' panel libraries
Ability to incorporate an actual session record into a fresh simulation to build demonstrations and on-line training courses
Easy synchronization of a voice/keystroke tape and a host computer system
Reliable identification (subject, date, experimental condition) of recorded data
Simulation flow definition corresponds naturally to a handwritten graphic format, e.g., a man-machine function diagram
Quickly picked up by temporary staff (There is a heavy workload in building and maintaining a simulation, which represents a poor use of highly trained professional human factors staff.)
Easy for human factors professionals not familiar with data processing to understand a simulation they did not write, to suggest modifications, and to run it in an experiment
Resists crashes during an experiment, even if incomplete
Easy to restart a session if forced to suspend it
Successive versions easily archived and restored for examination
Easy to manage several current versions of the same simulation at once
Easily transmitted to other computer installations
Easy to load and examine a simulation received from another location
Easily used to assist detailed dialogues with designers and reviewers

The ICU simulation was written, as stated earlier, using an informal system based on the VM/CMS EXEC interpreter and a high-level screen handler called IOS3270.[3] As a result of this and other experiments with this apparatus, a clear idea was obtained of its shortcomings. In Table 3 we reproduce a list of the main points to consider when choosing a simulation tool for this sort of work.

A simulation tool called SIMIC (System Intended to Mimic Interactive Conversation)[18] has been written to accommodate most if not all of these requirements. It runs under VSAPL for CMS[19] and also makes use of IOS3270.[3]

Summary

A design exercise that resulted in a draft set of panels for the on-line HELP facility of the Interactive Chart Utility (ICU) has been described. The exercise was undertaken before the program code was completed for the product and suggested certain modifications to the latter.

The exercise entailed running trials in which pairs of subjects not familiar with data processing used a simulation of the ICU. Their keyboard activity and utterances were recorded in synchrony, from which a faithful action replay was possible. This replay assisted the drafting of a suitable set of HELP panels by the experimenter.

The validity of running cut-down behavioral investigations as part of a design project was discussed, as opposed to procedures in scientific research, where such experiments would have little validity. Three simple objective measures of the success of the drafting exercise were explored, of which the most robust under practical conditions appears to be one based on the kinds of recurrent error that subjects make.

The feasibility of this sort of exercise is critically dependent on the quality of the simulation tool used. A list of requirements was stated, based upon the experience of this and similar studies.

ACKNOWLEDGMENTS

My gratitude goes to Dr. Ben Shneiderman for his timely encouragement and to the referees for their advice on the most appropriate presentation of this material; to Dr. Nick Hammond, Dr. Phil Barnard, and Dr. John Morton of the MRC Applied Psychology Unit, Cambridge, England, and Dr. John Long, Director of Studies at the Ergonomics Unit, University College, London, for stimulating discussions and for laying the technical foundations for this work; to the GDDM design team at Hursley for their cooperation and forbearance; and to my various students, especially Charles McAndrew and Paul Kingsnorth, who have patiently built simulations and analyzed records to make this exercise possible.

CITED REFERENCES

1. *IBM Graphical Data Display Manager (GDDM) and Presentation Graphics Feature (PGF): General Information*, GC33-0100, Program Number 5748-XXH, IBM Corporation (1980); available through IBM branch offices.
2. *IBM Presentation Graphics Feature Interactive Chart Utility, Introductory Course*, SC33-0111, Program Number 5748-XXH, IBM Corporation (1980); available through IBM branch offices.
3. *IBM Display Input/Output Facility, Program Description and Operations Manual*, SB11-5329, Program Number 5785-HAA, IBM Corporation (1979); (The display module is named IOS3270.); available through IBM branch offices.
4. N. Hammond, J. Long, I. Clark, P. Barnard, and J. Morton, "Documenting human-computer mismatch in interactive systems," *9th International Symposium on Human Factors in Telecommunications*, Holmdel, NJ (1980).
5. A. Newell and H. A. Simon, *Human Problem Solving*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1972).
6. E. M. Johnson and J. D. Baker, "Field testing: the delicate compromise," *Human Factors* **16**, No. 3, 203–214 (1974).
7. A. Chapanis, "The relevance of laboratory studies to practical situations," *Ergonomics* **10**, No. 5, 557–577 (1967).

8. E. C. Poulton, "Observer bias," *Applied Ergonomics* **6**, No. 1, 3–8 (1975).
9. L. S. Hearnshaw, *Cyril Burt: Psychologist*, Hodder & Stoughton, London (1979).
10. N. Hammond, P. Barnard, I. Clark, J. Morton, and J. Long, "Structure and content in interactive dialogue," *88th Annual Convention of American Psychological Association*, Montreal, Canada (1980).
11. W. L. Hays, *Statistics*, Holt, Rinehart, and Winston, New York (1969), pp. 332–333.
12. H. Sackman, *Experimental Investigation of User Performance in Time-Shared Computing Systems: Retrospect, Prospect, and the Public Interest*, Report No. SP-2846, System Development Corporation, Santa Monica, CA (1967).
13. J. Morton, P. J. Barnard, N. V. Hammond, and J. B. Long, "Interacting with the computer: A framework," *Teleinformatics '79*, E. J. Boutmy and A. Dantine (Editors), North-Holland Publishing Co., Amsterdam, pp. 201–208.
14. I. A. Clark, *How to Help "HELP" Help*, Human Factors Laboratory Report HF 022, IBM Corporation, Hursley, England (1980).
15. R. Gunning, *How to Take the Fog Out of Writing*, Dartnell Press, Inc., Chicago (1959), pp. 9–10.
16. I. A. Clark, "Human factors in designing software prototypes," *Proceedings of DESIGN '79 Symposium* (Monterey, CA) **1**, 333–346 (April 1979).
17. E. Meijer, "Application simulation," *Proceedings of DESIGN '79 Symposium* (Monterey, CA) **1**, 410–420 (April 1979).
18. I. A. Clark and P. Kingsnorth, *SIMIC, a Simulation Tool for Human Factors in Product Development*, Human Factors Laboratory Report HF 033, IBM Corporation, Hursley, England (1980). This tool is not available outside IBM.
19. *IBM VSAPL for CMS: Terminal User's Guide*, SH20-9067, Program Number 5748-AP1, IBM Corporation (1976); available through IBM branch offices.

The author is located at the IBM United Kingdom Laboratories Limited, Hursley Park, Winchester, Hampshire SO21 2JN, England.

Reprint Order No. G321-5149.