Human Factors Testing in the Design of Xerox's 8010 "Star" Office Workstation

William L. Bewley, Teresa L. Roberts, David Schroit, William L. Verplank

Xerox Office Systems Division

Abstract

Integral to the design process of the Xerox 8010 "Star" workstation was constant concern for the user interface. The design was driven by principles of human cognition. Prototyping of ideas, paper-and-pencil analyses, and human-factors experiments with potential users all aided in making design decisions. Three of the human-factors experiments are described in this paper: A *selection schemes* test determined the number of buttons on the mouse pointing device and the meanings of these buttons for doing text selection. An *icon* test showed us the significant parameters in the shapes of objects on the display screen. A *graphics* test evaluated the user interface for making line drawings, and resulted in a redesign of that interface.

1. Introduction

The Xerox 8010 office workstation, known as Star during development, is meant for use by office professionals. In contrast to word processors which are largely used by secretarial and administrative personnel, or computer systems which are largely used by technically-trained workers, Star had to be designed for casual users who demand extensive functionality at a small training cost. Since the background of the targeted users was very different from that of Star's designers, the designers' intuitions could not always be used as the criteria for an acceptable system.

Recognizing that design of the Star user interface was a major undertaking, the design team approached it using several principles, derived from cognitive psychology:

- There should be an explicit user's model of the system, and it should be familiar (drawing on objects and activities the user already works with) and consistent.
- Seeing something and pointing to it is easier for people than remembering a name and typing it. This principle is often expressed in psychological literature as "recognition is generally easier than recall" [Anderson].
- Commands should be uniform across domains, in cases where the domains have corresponding actions (*e.g.*, deleting a word from text, deleting a line from an illustration, and deleting information from a database).
- The screen should faithfully show the state of the object the user is working on: "What you see is what you get."

Even given these principles, the design space is enormous, and many proposed designs turned out to be unsatisfactory. Further tools were needed for designing Star than just a set of principles to start from. Once a design was proposed, it had to be tested, which we did in several ways.

First, the general user interface was prototyped in an environment which made it easy to modify. Care was spent on the user illusion, but not on all the underpinnings necessary to provide an integrated, robust system. This prototype was used by Star designers and others to get a "feel" for what they were proposing.

Sometimes a prototype was not appropriate to answer questions arising in the design, so various analyses were performed. For instance, Card, Moran, and Newell's Keystroke Level Model [Card] was used to study the number of user actions and amount of time required to perform large office tasks, given a proposed command language. This helped identify bottlenecks and annoyances in the procedures that would be necessary to perform the tasks.

Finally, in certain domains where neither analysis nor informal use of prototypes was sufficient to validate or invalidate proposed designs, the Functional Test Group (which also did much of the user interface analysis) performed formal human-factors experiments. Those experiments are the topic of this paper.

In the rest of the paper, we first present the basics of the Star user interface, to give the reader the context of the tests which were run. Then we describe three representative experiments which we performed. Finally, we discuss what sort of things were tested successfully and what sort of things were not tested, significant features of the testing we did, and the effect the testing had on the success of Star's user interface.

2. Background description of Star

The Star user interface has been extensively described in papers which also address the design philosophy and process [Seybold, Smith1, and Smith2]. Here we describe only enough of Star to motivate the user interface tests we will be covering.

Star is run on a powerful personal computer. It has a 17" diagonal, high-resolution, bitmapped screen which can display arbitrarily complex images; a keyboard which has a moderate number of function keys to the left, right, and above the main typing array; and a pointing device (the mouse). Figure 1 shows these elements graphically.

Central to the user interface is the office metaphor. Familiar office objects, such as documents, folders, and file drawers, are represented on the screen by small pictures called *icons*. Data icons, such as documents, are mailed, filed, and printed by moving them to icons representing outbaskets, file drawers, and printers, respectively, so individual commands are not needed for these operations. When the content of an object needs to be seen, such as for editing, the icon is *opened* to take up a large rectangular area.

on the screen called a *window*.

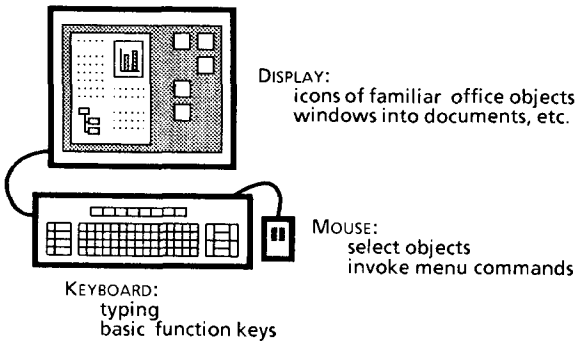

Figure 1. Elements of the Star Workstation

Star documents include text, graphics, typeset mathematical formulas, and tables, all freely intermixed. All appear on the screen exactly as they will appear when they are printed (within the limits of the display resolution), and all can be edited interactively.

The user performs a Star action by first selecting the object of the action by pointing to it with the mouse; it videoinverts to give feedback that it is selected. After making a selection, the user presses the function key indicating the desired command. Most Star actions can be performed with only four function keys: Delete, Move, Copy, and Show Properties. These are applied to all kinds of Star objects from characters and paragraphs to data-driven barcharts and icons. The function of Delete is clear. Move and Copy, in addition to allowing rearrangement and replication of objects, perform printing, mailing, and filing functions, as mentioned above.

The Show Properties key brings up a *property sheet*. Each Star object has a set of properties displayed on its property sheet. For instance, the properties of a character are its typeface, size, position with respect to the baseline, and so forth. The properties of a folder (a collection of documents and other folders) include its name and the sort order of its contents. The properties of a data-driven barchart include information on the desired orientation and shading of the bars, the number of ticks on the axis, and, of course, the data. The property sheets appear when asked for, let the user select desired property settings, and then disappear when no longer needed. They offer an immense flexibility of options for Star objects, without cluttering either a command language or the screen.

3. Selection Schemes Tests

The goal of the two selection schemes tests was to evaluate methods for selecting text. The schemes are various mappings of one, two, or three mouse buttons to the functions needed for indicating what text is to be operated on. The kinds of selection behavior needed are (1) *Point*: indicating a point between two characters, to be used as the destination of a Move or Copy, or the position where new typed text will be inserted; (2) *Select*: selecting some text, possibly in increments of a character, word, sentence, paragraph, or the whole document; and (3) *Extend*: extending the selection to include a whole range of text.

Selection Scheme Test 1

The first test compared six selection schemes. These schemes are summarized in Figure 2, schemes A through F. The six selection schemes differ in the mapping between mouse buttons and the three operations. As one example of the differences among schemes, in two schemes, A and B, different buttons are used for Point and Select, while in the remaining four schemes the first button is used for both Point and Select.

Methodology. Using a between-subjects paradigm, each of six groups (four subjects per group) was assigned one of the six schemes. Two of the subjects in each group were experienced in the use of the mouse, two were not. Each subject was first trained in the use of the mouse and in basic Star editing techniques. Next, the assigned scheme was taught. Each subject then performed ten text editing tasks, each of which was repeated six times. Dependent variables were selection time and selection errors.

Selection time. Mean selection times are shown in Figure 3. Among these six schemes, scheme F was substantially better than the others over all six trials ($p < .001$).

	Scheme A	Scheme B	Scheme C	Scheme D	Scheme E	Scheme F	Scheme G
Button 1	Point	Point	Point C Drawthrough	Point C, W, S, ¶, D Drawthrough	Point C, W, S, ¶, D Drawthrough	Point C Drawthrough	Point C, W, S, ¶, D
Button 2	C Drawthrough	C, W, S, ¶, D Drawthrough	W, S, ¶, D Drawthrough		Adjust	Adjust	Adjust
Button 3	W, S, ¶, D Drawthrough						

Key: Point: Selects a point, *i.e.*, a position between adjacent characters. Used as destination for Move or Copy. If the button doesn't also make a text selection, Point is also used to indicate a destination for type-in.

C, W, S, ¶, D: Selects a character, word, sentence, paragraph, or whole document, by repeatedly clicking the mouse button while pointing at something that's already selected.

Drawthrough: The user holds the button down and moves the mouse. The selection extends from the button-down position to the button-up point. The selection is extended in units of whatever was previously selected.

Adjust: The user clicks the mouse button to extend the selection from the existing selection to the button-up point. The selection is extended in units of whatever was previously selected.

Figure 2. Description of the Selection Schemes

Scheme A	Scheme B	Scheme C	Scheme D	Scheme E	Scheme F	Scheme G
12.25	15.19	13.41	13.44	12.85	9.89	7.96

Figure 3. Mean Selection Time (Secs)

Selection Errors. There was an average of one selection error per 4 tasks. The majority (65%) were errors in drawthrough: either too far or not far enough. The frequency of drawthrough errors did not vary as a function of selection scheme. "Too Many Clicks" errors, *e.g.*, the subject clicking to a sentence instead of a word, accounted for 20% of the errors, with schemes which employed less multiple-clicking being better. "Click Wrong Mouse Button" errors accounted for 15% of total errors. These errors also varied across schemes, with schemes having fewer buttons being better.

Selection Scheme Test 2

The results of the first test were interpreted as suggesting that the following features of a selection scheme should be avoided: 1) drawthrough, 2) three buttons, and 3) multiple-clicking. The second selection scheme test evaluated a scheme designed with these results in mind. Scheme G is also described in Figure 2. It is essentially Scheme F with the addition of multiple-clicking. It avoids drawthrough and uses only two buttons. Multiple-clicking is used because, although 20% of the errors in the first test were attributable to errors in multiple-clicking, Star's designers felt that a selection scheme must provide for quick selection of standard text units.

The same methodology was used for evaluating the new scheme as was used for the rest, except that only one user was experienced with the mouse and three were not.

Results. The mean selection time for the new scheme was 7.96 sec, the lowest time so far. The frequency of "Too Many Clicks" errors in Scheme G was about the same as the frequency observed in the first selection scheme test.

Conclusions. The results of the second test were interpreted as indicating that Scheme G was acceptable for use in Star, since (1) selection time for Scheme G was shorter than for all other schemes, and (2) the advantage of providing quick selection of standard text units through multiple-clicking was judged sufficiently great to balance the moderate error rate due to multiple-clicking errors.

4. Icon Shape Test

A series of tests was used in helping to decide what the icons should look like so that they would be readily identifiable, easy to learn, and distinguishable. The purpose of the tests was to give some feedback to the icon designers about probable user response to designs. We did not intend that the tests alone be used to decide which set of icons was best, but rather to point up difficulties and preferable design directions.

We did not test icons as commands. These tests did not consider the issues of whether iconic representation and implicit commands are better than typed names and typed commands or whether a small set of "universal" commands (Delete, Move, Copy, Show Properties) applied uniformly across domains (text, graphics, printing, mailing) are superior to a large number of commands specialized to each domain.

Methodology and Results

Four different sets of 17 icons were designed by four different designers (see Figure 4). Five subjects were assigned to each set for a total of 20 subjects. A series of paper-and-pencil tests was used to assess familiarity (Naming Tests); two response-time tests using a computer and display measured recognizability and distinguishability (Timed Tests); finally, subjects were asked for their opinions (Rating Tests).

Naming Tests. First the experimenter showed the icons one at a time, each on a 3×5" card, and asked for "a short description of what you think it is." Then the entire set was presented and the subjects were allowed to change their descriptions. Next, names and short descriptions were given and the subject was asked to "point to the symbol that best fits each description." Finally, with all the names available, the subject was asked to put "one of the names next to each symbol."

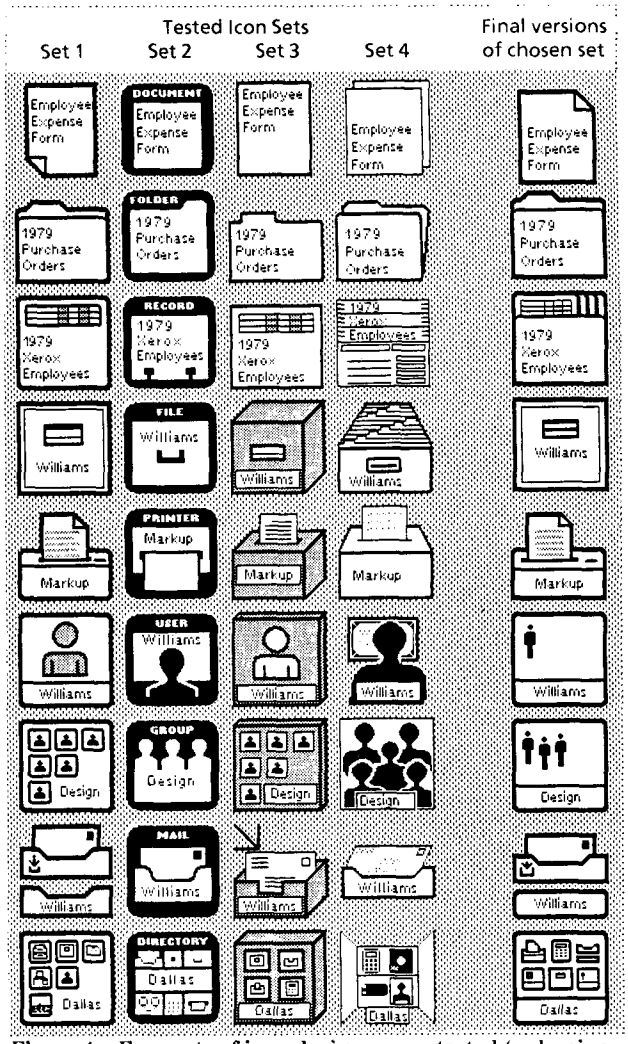

Figure 4. Four sets of icon designs were tested (only nine of the seventeen in each set are shown here). Set 1 was chosen and modified as shown at the right.

Since Set 2 had each icon named already, the naming tests showed the obvious value of having labels on icons. The three sets without labels were misinterpreted about 25% of the time on first sight. A few specific icons were revealed as most difficult: Printer (Sets 3 and 4), Record File (1, 3, 4), Directory (3, 4), Group (1, 3). For example, the Group from Set 1 was described as "cemetery plots -- to purge information" and as "keyboard -- pushbuttons".

Timed Tests. The two timed tests used a Xerox Alto computer with the icons displayed on the screen as they would be in Star. For the first timed test, we used a procedure suggested by Pew and Green [Green]. The subjects were given the name of an icon and told that it may or may not appear on the display. When an icon appeared they responded as quickly as possible by pressing a YES- or a NO-button depending on whether they thought the one presented was the one named. This test showed no significant differences among the icon sets. We concluded that the short training involved in the Naming Tests was adequate for any of the sets.

In the second timed test, we asked the subjects to point as quickly as possible to the named icon in a randomized display of all the icons. Results of this test, combined with the naming results, are shown in Figure 5. This test showed some significant differences

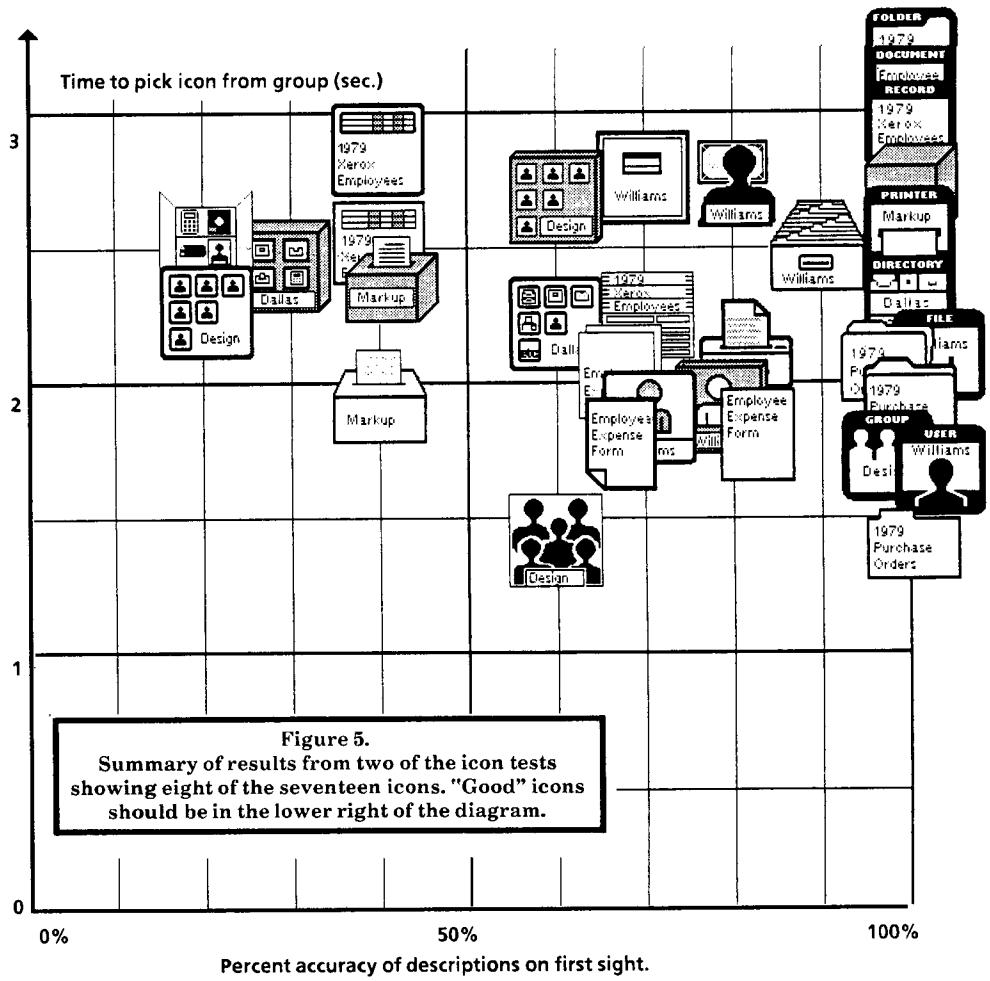

Figure 5.
Summary of results from two of the icon tests showing eight of the seventeen icons. "Good" icons should be in the lower right of the diagram.

among sets and icons. Over all, subjects with Set 2 took roughly 0.5 seconds longer than subjects with the other sets to find icons (2.5 vs 2.0), and subjects took more than a second longer to find the Document and Folder than to find the other icons (3.0 vs. 2.0).

Rating Tests. At the end of the tests, subjects were asked to say whether any of the icons in their set were "easy" or "difficult ... to pick out of the crowd". Subjects' opinions corresponded fairly well with their performance.

When shown all four sets and asked to choose a best icon for each type, subjects usually chose on the basis of which was most realistically depicted or because of the labels. Over-all preference was given to Set 2 ("most helpful") or to Set 4 ("more different shapes"). The opinions strongly reflect the tasks given in the tests; considerations beyond the tests would have been difficult for the subjects to judge.

Conclusions

The naming tests pointed out the value of labels (in Set 2), but the YES-NO response-time test indicated that, once learned, there was little difference among the sets for recognition. The pointing test, where distinguishability was important, showed that the sets with more visual variety (Sets 1, 3, and 4) were more successful. The most useful results from the icon tests were recommendations about specific icons; those with problems were redesigned.

The final choice of icon designs included a variety of concerns beyond those that could be addressed by the tests. For example, to give the user feedback that a particular icon is selected, its image is inverted (everything white becomes black and vice versa). Set 1, which has every icon predominantly white, was considered the best at showing selection. Finally, an important consideration in choosing the icon designs was how refined the set was graphically. With some redesign, Set 1 was the final choice for Star.

5. Graphics Tests

Unlike the two tests just described, the goal of the graphics testing was much less clearcut. We simply wanted to find out how easy the user interface was to learn, and where the difficulties were.

The Star graphics functionality, described in detail in [Lipkie], involves a structured graphics approach to making line drawings. Lines and rectangles, like other Star objects such as icons and characters, are objects that can be selected, moved, copied, and altered. According to the original user interface at the time of the tests, selection of graphics objects followed the text paradigm closely (see Figure 6): clicking the left mouse button once at an object (such as a line) selected one point on the object (an end of the line); a second click of the left button enlarged the selection so that it included the whole object. Because of this richness in selecting, few function keys were able to perform many functions. For instance, a user could lengthen, shorten, and rotate ("stretch") a line by selecting only one end and pressing the Move key. The same key moved the entire line if the whole line were selected (by clicking twice). Creation of new lines was done with the Copy key, with a special accelerator when only an end of a line was selected that aided in making connected lines. Captions could be added to the illustration by copying into the picture a "text frame," a rectangular area which was capable of containing text. Prototype examples of all graphics objects could be obtained from a system-supplied document called a "graphics transfer sheet."

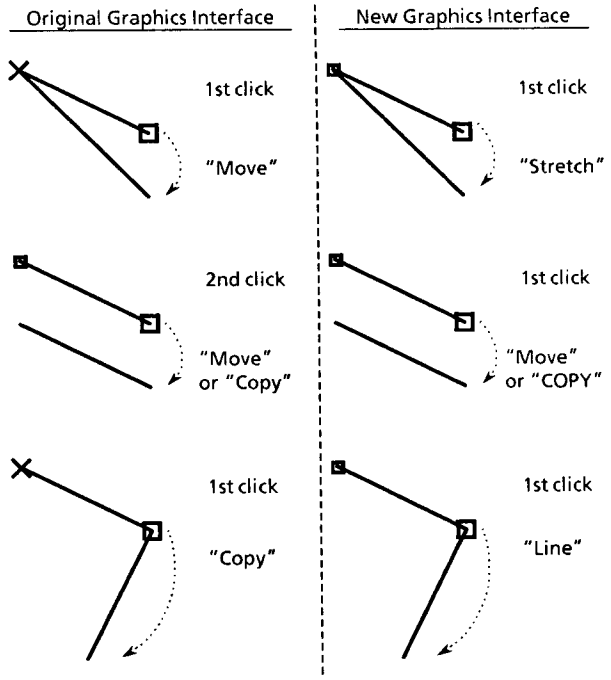

Figure 6. Graphic selections and commands. The new scheme simplified selection by eliminating multiple-clicking and adding graphics-only commands (Stretch and Line).

Methodology

This experiment used a small number (3) of inexperienced subjects, since we were looking for qualitative behavior, rather than statistical significance in these tests. The subjects had already been through a prototype of Star's on-line introduction to the general functions, so they had a background in the use of Star, and we knew roughly how they fit into the spectrum of Star users. For this test the subjects read hardcopy graphics training which consisted of explanatory material, interspersed with exercises done on the machine. At the end of the training, the subjects were asked to create some illustrations, both from scratch and by modifying existing illustrations. The test was self-paced. Time and performance were the dependent variables.

About five weeks later, two of the three subjects returned to do some exercises to show how much of the training they had retained.

The entire study (taking up to one day for the test, and one hour for the follow-up) was videotaped. Cameras showed both the user and the screen, along with the time of day.

Results

Both during the test and follow-up, evaluators recorded the times spent in each part of the training and exercises, plus critical incidents in the use of the system. These critical incidents were later catalogued into problems with the prototype implementation, with the design of the user interface, with the training, and with the design of the experiment. They were also prioritized according to how pervasive and persistent they were.

The design problems were described to the Star design group, and were reinforced by showing the designers clips of the videotape. There were two major user-interface problems: First, the multiple clicking that distinguished selection of the end of an object from selection of the whole object was far too error-prone. Selection should be made at one level only. This necessitated addition of a function key for the Stretch function, since the Move key could no longer do double duty. Secondly, the Copy method of making a new line was too awkward. Since making a new line is central to graphics, it was felt that a function key should be dedicated to this operation.

Redesign. Both of these changes to the user interface involved adding new function keys. But at that time the number of keys on the Star keyboard was frozen, and all had assigned meanings. The suggested solution was to change the meanings of the function keys across the top of the keyboard, since (a) they were already being changed in another context, and (b) they were normally just accelerators for text functions and had no use in the graphics context. The new meanings of the keys would be displayed on the screen whenever they were in effect. There were eight keys there, but only two were needed to solve the problems found by testing. However, the inventive designers quickly found uses for most of the rest.

After this redesign, the graphics user interface was presumably easier to use. But the new design added complication to Star in general by allowing function keys to change their meaning in a way much more obtrusive than before. We did not know whether the overall effect was an improvement or not, so the test was repeated to compare the new scheme with the old.

Retest. The second graphics test fixed several problems in the experiment design, and used early versions of the customer training materials. It was run similarly to the first, with three subjects learning the old user interface and four learning the new one. The results of the repeated test of the old user interface were very similar to those of the original test. Both versions took similar amounts of time in the training portion, but at the end the users of the new interface were quicker at making illustrations and finished more of the tasks (see Figure 7). New problems were identified, of course, but they were relatively easy to fix, so the new user interface was the one adopted for the product.

	Old Interface	New Interface
Time per training module (min.)	32 ± 12	42 ± 12
Time per task (min.)	18 ± 5	9 ± 5

Numbers are given as $M \pm SD$, where M is the mean over all the users and SD is the standard deviation.

Figure 7. Quantitative Comparison of Graphics Interfaces

6. Summary and Conclusions

The three experiments described here run the gamut from formality to informality, depending on the purposes of the tests and the costs of the experiments. In general, we were able to be most formal and careful when the topic of the experiment was well-defined and when the experiments could be kept short. As the questions to be settled became less well-defined, on the other hand, experiments took on a flavor of "fishing expeditions" to see what we came up with. Particularly when we addressed problems relating to use of a general Star function and the relationship of that function to the rest of Star, the experiments required large amounts of training. This was very costly both in setting up the tests and in execution; a consequence was that fewer subjects were used. Finally, extremely vague questions, such as whether icons in general provide a better user interface than typing commands, were not tested at all; icons were shown to be an acceptable user interface, and that result sufficed for our purposes.

Important points we found in our experimentation are the following:

(1) Videotaping was a very important tool. First of all, the cameras allowed us to see and hear the subject without being in the same room. Secondly, it was a record of all activity, so we didn't need to take perfect notes during the

experiment. Third, the designers were more convinced by the videotapes than by our dry numbers that people were having trouble with their system.

(2) All tests were flexible enough to allow the experimenters to observe why results were coming out the way they were. For example, verbal protocols were elicited in many of the tests and formal or informal interviews followed all the tests. This was important in helping us suggest design improvements.

Star was a mammoth undertaking. "The design effort took more than six years. ... The actual implementation involved from 20 to, eventually, 45 programmers over 3.5 years producing over 250,000 lines of highlevel code." [Harslem] By the time of the initial Star release, the Functional Test Group had performed over 15 distinct human-factors tests, using over 200 experimental subjects and lasting for over 400 hours (Figure 8). In addition, we applied a standard methodology to compare Star's text editing features to those of other systems [Roberts]. The group averaged 6 people (1 manager, 3 scientists, and 2 assistants) for about 3 years to perform this work.

The impact of Functional Testing on the Star product has been a pervasive set of small and large changes to the user interface. The amount of difference these changes made is, of course, impossible to assess, but the quality of Star's user interface is well known. It has won an award as the "friendliest" computer system of 1982, as judged by *Computing* magazine. Imitators, led by Sidereal, Apple's Lisa, and VisiCorp's Visi^{ON}, are starting to have a major impact on the marketplace. We can only take this as a ratification of Star's design process, a rich blend of user interface principles, functional analysis, and human interface testing.

Test Topic	No. Sub	Tot. Hrs	Impact
Selection Schemes	28	64	Lead to new design; validated new scheme
Keyboard (6 layouts)	20	40	Led to design of keyboard
Display	20	10	Specified display phosphor and refresh rate
Tab-indent	16	16	Caused redesign of Tab and Indent functionality
Labels	12	6	Caused change in property sheet and keyboard labels
Property Sheets	20	40	Identified potential interface problems and redesigns
Fonts	8	6	Led to decision on screen-paper coordination
Icons	20	30	Led to design of icons
Initial Dialogue	12	36	Led to design of training facility and materials
HELP	2	6	Validated HELP design ideas
Graphics	10	65	Led to redesign; validated new design
Graphic Idioms	4	16	Contributed to redesigns
J-Star Labels	25	25	Led to design of keyboard labels for Japanese-Star

Figure 8. Partial listing of Star-1 Functional Tests

Acknowledgments

The tests described here were carried out by the staff of the Functional Test Group consisting of W. Bewley, C. McBain, L. Miller, T. Roberts, D. Schroit, W. Verplank, and R. Walden. Able assistance was provided by M. Beard, W. Bowman, N. Cox, A. Duvall, W. Judd, J. Newlin and D. Silva. Star user-interface design was the result of a long process of innovation at Xerox PARC and elsewhere; however immediate credit should go to Eric Harslem, Charles Irby, Ralph Kimball, and David C. Smith.

References

[Anderson] Anderson, J. R. *Cognitive psychology and its implications.* W. H. Freeman and Company, San Francisco, 1980.

[Card] Card, S. K., Moran, T. P., and Newell, A. The Keystroke-Level Model for user performance time with interactive systems. *Communications of the ACM, 23,* 7, (July 1980), 396-410.

[Green] Green, P. and Pew, R. W. Evaluating pictographic symbols: an automotive application. *Human Factors, 20,* 1, (Feb. 1978), 102-114.

[Harslem] Harslem, E., Nelson, L. E. A retrospective on the development of Star. *Proc. of the 6th International Conference on Software Engineering,* Tokyo, Japan, (Sept. 1982), 377-383.

[Lipkie] Lipkie, D. E., Evans, S. R., Newlin, J. K., Weissman, R. L. Star graphics: an object-oriented implementation. *Computer Graphics, 16,* 3, (July 1982), 115-124.

[Roberts] Roberts, T. L. and Moran, T. P. The evaluation of text editors: methodology and empirical results. *Communications of the ACM, 26,* 4, (April 1983)

[Seybold] Seybold, J. W. The Xerox Star, a "professional" workstation. *The Seybold Report on Word Processing, 4,* 5, (May 1981), 1-19.

[Smith1] Smith, D. C., Irby, C., Kimball, R., Verplank, W., Harslem, E. Designing the Star user interface. *Byte, 7,* 4, (April 1982), 242-282.

[Smith2] Smith, D. C., Harslem, E., Irby, C., Kimball, R. The Star user interface: an overview. *Proceedings of the AFIPS 1982 National Computer Conference, 50,* (June 1982), 515-528.