The star user interface: an overview

by DAVID CANFIELD SMITH, CHARLES IRBY, and RALPH KIMBALL
Xerox Corporation
Palo Alto, California
and
ERIC HARSLEM
Xerox Corporation
El Segundo, California

ABSTRACT

In April 1981 Xerox announced the 8010 Star Information System, a new personal computer designed for office professionals who create, analyze, and distribute information. The Star user interface differs from that of other office computer systems by its emphasis on graphics, its adherence to a metaphor of a physical office, and its rigorous application of a small set of design principles. The graphic imagery reduces the amount of typing and remembering required to operate the system. The office metaphor makes the system seem familiar and friendly; it reduces the alien feel that many computer systems have. The design principles unify the nearly two dozen functional areas of Star, increasing the coherence of the system and allowing user experience in one area to apply in others.

INTRODUCTION

In this paper we present the features in the Star system without justifying them in detail. In a companion paper,[1] we discuss the rationale for the design decisions made in Star. We assume that the reader has a general familiarity with computer text editors, but no familiarity with Star.

The Star hardware consists of a processor, a two-page-wide bit-mapped display, a keyboard, and a cursor control device. The Star software addresses about two dozen functional areas of the office, encompassing document creation; data processing; and electronic filing, mailing, and printing. Document creation includes text editing and formatting, graphics editing, mathematical formula editing, and page layout. Data processing deals with homogeneous databases that can be sorted, filtered, and formatted under user control. Filing is an example of a network service using the Ethernet local area network.[2,3] Files may be stored on a work station's disk (Figure 1), on a file server on the work station's network, or on a file server on a different network. Mailing permits users of work stations to communicate with one another. Printing uses laser-driven xerographic printers capable of printing both text and graphics. The term *Star* refers to the total system, hardware plus software.

As Jonathan Seybold has written, "This is a very different product: Different because it truly bridges word processing and typesetting functions; different because it has a broader range of capabilities than anything which has preceded it; and different because it introduces to the commercial market radically new concepts in human engineering."[4]

Figure 1—A Star workstation showing the processor, display, keyboard and mouse

The Star hardware was modeled after the experimental Alto computer developed at the Xerox Palo Alto Research Center.[5] Like Alto, Star consists of a Xerox-developed high-bandwidth MSI processor, local disk storage, a bit-mapped display screen having a 72-dot-per-inch resolution, a pointing device called the mouse, and a connection to the Ethernet. Stars are higher-performance machines than Altos, being about three times as fast, having 512K bytes of main memory (vs. 256K bytes on most Altos), 10 or 29M bytes of disk memory (vs. 2.5M bytes), a 10½-by-13½-inch display screen (vs. a 10½-by-82-inch one), 1024×808 addressable screen dots (vs. 606×808), and a 10M bits-per-second Ethernet (vs. 3M bits). Typically, Stars, like Altos, are linked via Ethernets to each other and to shared file, mail, and print servers. Communication servers connect Ethernets to one another either directly or over phone lines, enabling internetwork communication to take place. This means, for example, that from the user's perspective it is no harder to retrieve a file from a file server across the country than from a local one.

Unlike the Alto, however, the Star user interface was designed before the hardware or software was built. Alto software, of which there was eventually a large amount, was developed by independent research teams and individuals. There was little or no coordination among projects as each pursued its own goals. This was acceptable and even desirable in a research environment producing experimental software. But it presented the Star designers with the challenge of synthesizing the various interfaces into a single, coherent, uniform one.

ESSENTIAL HARDWARE

Before describing Star's user interface, we should point out that there are several aspects of the Star (and Alto) architecture that are essential to it. Without these elements, it would have been impossible to design a user interface anything like the present one.

Display

Both Star and Alto devote a portion of main memory to the bit-mapped display screen: 100K bytes in Star, 50K bytes (usually) in Alto. Every screen dot can be individually turned on or off by setting or resetting the corresponding bit in memory. This gives both systems substantial ability to portray graphic images.

Memory Bandwidth

Both Star and Alto have a high memory bandwidth—about 50 MHz, in Star. The entire Star screen is repainted from memory 39 times per second. This 50-MHz video rate would swamp most computer memories, and in fact refreshing the screen takes about 60% of the Alto's memory bandwidth. However, Star's memory is double-ported; therefore, refreshing the display does not appreciably slow down CPU memory access. Star also has separate logic devoted solely to refreshing the display.

Microcoded Personal Computer

Both Star and Alto are personal computers, one user per machine. Therefore the needed memory access and CPU cycles are consistently available. Special microcode has been written to assist in changing the contents of memory quickly, permitting a variety of screen processing that would otherwise not be practical.[6]

Mouse

Both Star and the Alto use a pointing device called the mouse (Figure 2). First developed at SRI,[7] Xerox's version has a ball on the bottom that turns as the mouse slides over a flat surface such as a table. Electronics sense the ball rotation and guide a cursor on the screen in corresponding motions. The mouse is a "Fitts's law" device: that is, after some practice you can point with a mouse as quickly and easily as you can with the tip of your finger. The limitations on pointing speed are those inherent in the human nervous system.[8,9] The mouse has buttons on top that can be sensed under program control. The buttons let you point to and interact with objects on the screen in a variety of ways.

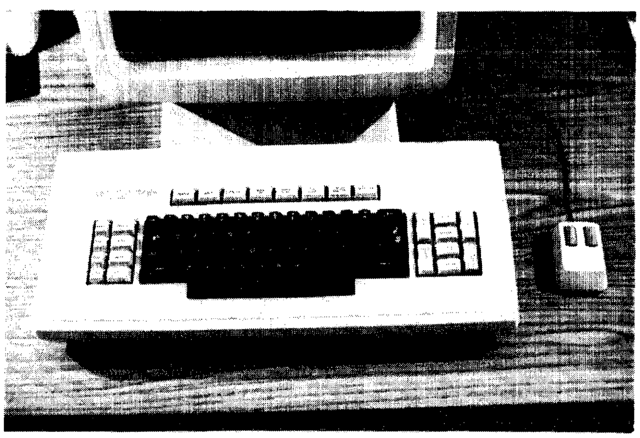

Figure 2—The Star keyboard and mouse

The keyboard has 24 easy-to-understand function keys. The mouse has two buttons on top.

Local Disk

Every Star and Alto has its own rigid disk for local storage of programs and data. Editing does not require using the network. This enhances the personal nature of the machines, resulting in consistent behavior regardless of how many other machines there are on the network or what anyone else is doing. Large programs can be written, using the disk for swapping.

Network

The Ethernet lets both Stars and Altos have a distributed architecture. Each machine is connected to an Ethernet. Other machines on the Ethernet are dedicated as *servers,* machines that are attached to a resource and that provide access to that resource. Typical servers are these:

1. *File server*—Sends and receives files over the network, storing them on its disks. A file server improves on a work station's rigid disk in several ways: (a) Its capacity is greater—up to 1.2 billion bytes. (b) It provides backup facilities. (c) It allows files to be shared among users. Files on a work station's disk are inaccessible to anyone else on the network.
2. *Mail server*—Accepts files over the network and distributes them to other machines on behalf of users, employing the Clearinghouse's database of names and addresses (see below).
3. *Print server*—Accepts print-format files over the network and prints them on the printer connected to it.
4. *Communication server*—Provides several services: The *Clearinghouse service* resolves symbolic names into network addresses.[10] The *Internetwork Routing service* manages the routing of information between networks over phone lines. The *Gateway service* allows word processors and dumb terminals to access network resources.

A network-based server architecture is economical, since many machines can share the resources. And it frees work stations for other tasks, since most server actions happen in the background. For example, while a print server is printing your document, you can edit another document or read your mail.

PHYSICAL OFFICE METAPHOR

We will briefly describe one of the most important principles that influenced the form of the Star user interface. The reader is referred to Smith et al.[1] for a detailed discussion of all the principles behind the Star design. The principle is to apply users' existing knowledge to the new situation of the computer. We decided to create electronic counterparts to the objects in an office: paper, folders, file cabinets, mail boxes, calculators, and so on—an electronic metaphor for the physical office. We hoped that this would make the electronic world seem more familiar and require less training. (Our initial experiences with users have confirmed this.) We further decided to make the electronic analogues be *concrete objects.*

Star documents are represented, not as file names on a disk, but as pictures on the display screen. They may be selected by pointing to them with the mouse and clicking one of the mouse buttons. Once selected, documents may be moved, copied, or deleted by pushing the MOVE, COPY, or DELETE key on the keyboard. Moving a document is the electronic equivalent of picking up a piece of paper and walking somewhere with it. To file a document, you move it to a picture of a file drawer, just as you take a piece of paper to a physical filing cabinet. To print a document, you move it to a picture of a printer, just as you take a piece of paper to a copying machine.

Though we want an analogy with the physical world for familiarity, we don't want to limit ourselves to its capabilities. One of the *raisons d'être* for Star is that physical objects do not provide people with enough power to manage the increasing complexity of their information. For example, we can take advantage of the computer's ability to search rapidly by providing a search function for its electronic file drawers, thus helping to solve the problem of lost files.

THE DESKTOP

Every user's initial view of Star is the Desktop, which resembles the top of an office desk, together with surrounding furniture and equipment. It represents a working environment, where current projects and accessible resources reside. On the screen (Figure 3) are displayed pictures of familiar office objects, such as documents, folders, file drawers, in-baskets, and out-baskets. These objects are displayed as small pictures, or *icons.*

You can "open" an icon by selecting it and pushing the OPEN key on the keyboard. When opened, an icon expands into a larger form called a *window,* which displays the icon's contents. This enables you to read documents, inspect the contents of folders and file drawers, see what mail has arrived, and perform other activities. Windows are the principal mechanism for displaying and manipulating information.

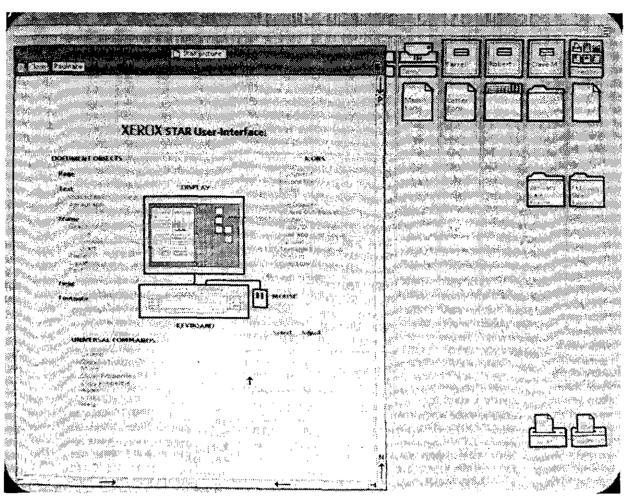

Figure 3—A "Desktop" as it appears on the Star screen

This one has several commonly used icons along the top, including documents to serve as "form pad" sources for letters, memos and blank paper. There is also an open window displaying a document.

The Desktop surface is displayed as a distinctive grey pattern. This is restful and makes the icons and windows on it stand out crisply, minimizing eye strain. The surface is organized as an array of 1-inch squares, 14 wide by 11 high. An icon may be placed in any square, giving a maximum of 154 icons. Star centers an icon in its square, making it easy to line up icons neatly. The Desktop always occupies the entire display screen; even when windows appear on the screen, the Desktop continues to exist "beneath" them.

The Desktop is the principal Star technique for realizing the physical office metaphor. The icons on it are visible, concrete embodiments of the corresponding physical objects. Star users are encouraged to think of the objects on the Desktop in physical terms. You can move the icons around to arrange your Desktop as you wish. (Messy Desktops are certainly possible, just as in real life.) You can leave documents on your Desktop indefinitely, just as on a real desk, or you can file them away.

ICONS

An *icon* is a pictorial representation of a Star object that can exist on the Desktop. On the Desktop, the size of an icon is approximately 1 inch square. Inside a window such as a folder window, the size of an icon is approximately ¼-inch square. Iconic images have played a role in human communication from cave paintings in prehistoric times to Egyptian hieroglyphics to religious symbols to modern corporate logos. Computer science has been slow to exploit the potential of visual imagery for presenting information, particularly abstract information. "Among [the] reasons are the lack of development of appropriate hardware and software for producing visual imagery easily and inexpensively; computer technology has been dominated by persons who seem to be happy with a simple, very limited alphabet of characters used to produce linear strings of symbols."[11] One of the authors has applied icons to an environment for writing programs; he found that they greatly facilitated human-computer communication.[12] Negroponte's Spatial Data Management system has effectively used iconic images in a research setting.[13] And there have been other efforts.[14,15,16] But Star is the first computer system designed for a mass market to employ icons methodically in its user interface. We do not claim that Star exploits visual communication to the ultimate extent; we do claim that Star's use of imagery is a significant improvement over traditional human-machine interfaces.

At the highest level the Star world is divided into two classes of icons, (1) data and (2) function icons:

Data Icons

Data icons (Figure 4) represent objects on which actions are performed. All data icons can be moved, copied, deleted, filed, mailed, printed, opened, closed, and have a variety of other operations performed on them. The three types of data icons are document, folder, and record file.

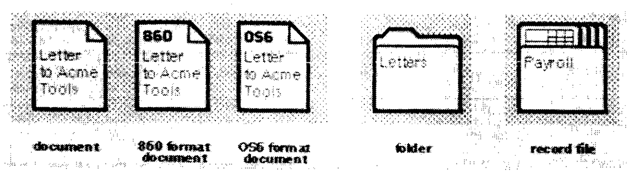

Figure 4—The "data" icons: document, folder and record file

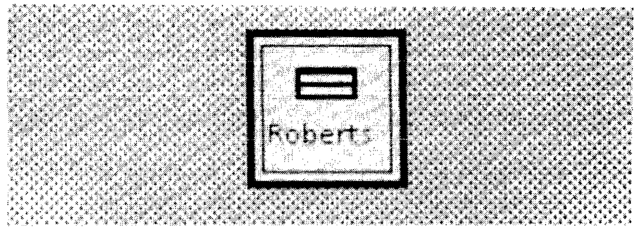

Figure 5—A file drawer icon

Document

A document is the fundamental object in Star. It corresponds to the standard notion of what a document should be. It most often contains text, but it may also include illustrations, mathematical formulas, tables, fields, footnotes, and formatting information. Like all data icons, documents can be shown on the screen, rendered on paper, sent to other people, stored on a file server or floppy disk, etc. When opened, documents are always rendered on the display screen exactly as they print on paper (informally called "what you see is what you get"), including displaying the correct type fonts, multiple columns, headings and footings, illustration placement, etc. Documents can reside in the system in a variety of formats (e.g., Xerox 860, IBM OS6), but they can be edited only in Star format. Conversion operations are provided to translate between the various formats.

Folder

A folder is used to group data icons together. It can contain documents, record files, and other folders. Folders can be nested inside folders to any level. Like file drawers (see below), folders can be sorted and searched.

Record file

A record file is a collection of information organized as a set of records. Frequently this information will be the variable data from forms. These records may be sorted, subset via pattern matching, and formatted into reports. Record files provide a rich set of information storage and retrieval functions.

Function Icons

Function icons represent objects that perform actions. Most function icons will operate on any data icon. There are many kinds of function icons, with more being added as the system evolves:

File drawer

A file drawer (Figure 5) is a place to store data icons. It is modeled after the drawers in office filing cabinets. The organization of a file drawer is up to you; it can vary from a simple list of documents to a multilevel hierarchy of folders containing other folders. File drawers are distinguished from other storage places (folders, floppy disks, and the Desktop) in that (1) icons placed in a file drawer are physically stored on a file server, and (2) the contents of file drawers can be shared by multiple users. File drawers have associated access rights to control the ability of people to look at and modify their contents (Figure 6).

Although the design of file drawers was motivated by their physical counterparts, they are a good example of why it is neither necessary nor desirable to stop with just duplicating real-world behavior. People have a lot of trouble finding things in filing cabinets. Their categorization schemes are frequently ad hoc and idiosyncratic. If the person who did the categorizing leaves the company, information may be permanently lost. Star improves on physical filing cabinets by taking advantage of the computer's ability to *search rapidly*. You can search the contents of a file drawer for an object having a certain name, or author, or creation date, or size, or a variety of other attributes. The search criteria can use fuzzy patterns containing match-anything symbols, ranges, and other predicates. You can also sort the contents on the basis of those criteria. The point is that whatever information retrieval facilities are available in a system should be applied to

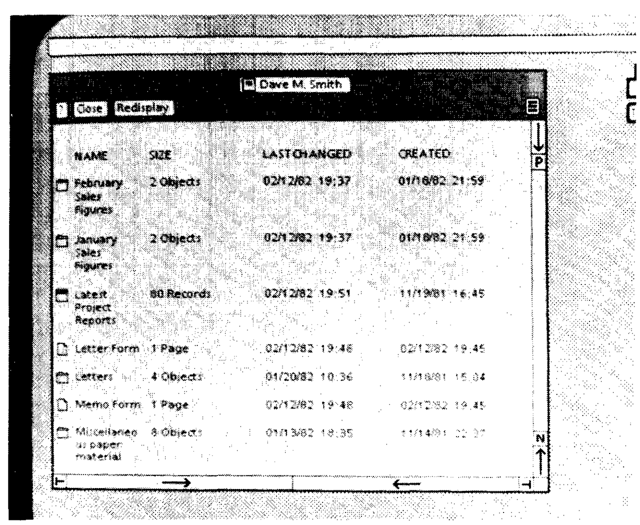

Figure 6—An open file drawer window

Note that there is a miniature icon for each object inside the file drawer.

the information in files. Any system that does not do so is not exploiting the full potential of the computer.

In basket and Out basket

These provide the principal mechanism for sending data icons to other people (Figure 7). A data icon placed in the Out basket will be sent over the Ethernet to a mail server (usually the same machine as a file server), thence to the mail servers of the recipients (which may be the same as the sender's), and thence to the In baskets of the recipients. When you have mail waiting for you, an envelope appears in your In basket icon. When you open your In basket, you can display and read the mail in the window.

Any document, record file, or folder can be mailed. Documents need not be limited to plain text, but can contain illustrations, mathematical formulas, and other nontext material. Folders can contain any number of items. Record files can be arbitrarily large and complex.

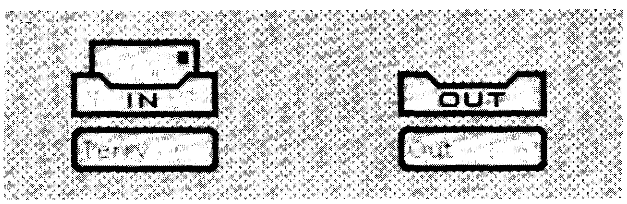

Figure 7—In and Out basket icons

Printer

Printer icons (Figure 8) provide access to printing services. The actual printer may be directly connected to your work station, or it may be attached to a print server connected to an Ethernet. You can have more than one printer icon on your Desktop, providing access to a variety of printing resources. Most printers are expected to be laser-driven raster-scan xerographic machines; these can render on paper anything that can be created on the screen. Low-cost typewriter-based printers are also available; these can render only text.

As with filing and mailing, the existence of the Ethernet greatly enhances the power of printing. The printer represented by an icon on your Desktop can be in the same room as your work station, in a different room, in a different building, in a different city, even in a different country. You perform exactly the same actions to print on any of them: Select a data icon, push the MOVE key, and indicate the printer icon as the destination.

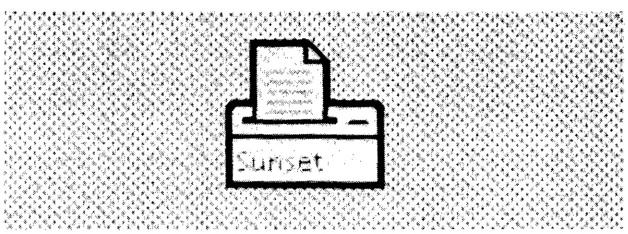

Figure 8—A printer icon

Floppy disk drive

The floppy disk drive icon (Figure 9) allows you to move data icons to and from a floppy disk inserted in the machine. This provides a way to store documents, record files and folders off line. When you open the floppy disk drive icon, Star reads the floppy disk and displays its contents in the window. Its window looks and acts just like a folder window: icons may be moved or copied in or out, or deleted. The only difference is the physical location of the data.

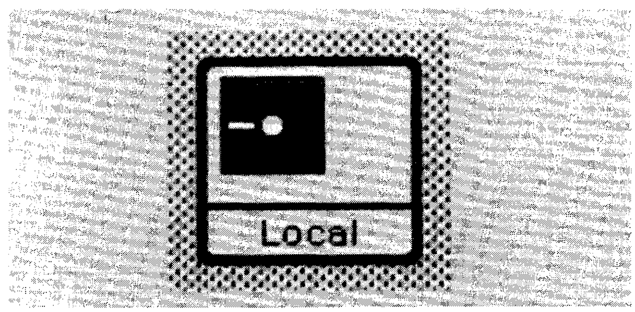

Figure 9—A floppy disk drive icon

User

The user icon (Figure 10) displays the information that the system knows about each user: name, location, password (invisible, of course), aliases if any, home file and mail servers, access level (ordinary user, system administrator, help/training writer), and so on. We expect the information stored for each user to increase as Star adds new functionality. User icons may be placed in address fields for electronic mail.

User icons are Star's solution to the naming problem. There is a crisis in computer naming of people, particularly in electronic mail addressing. The convention in most systems is to

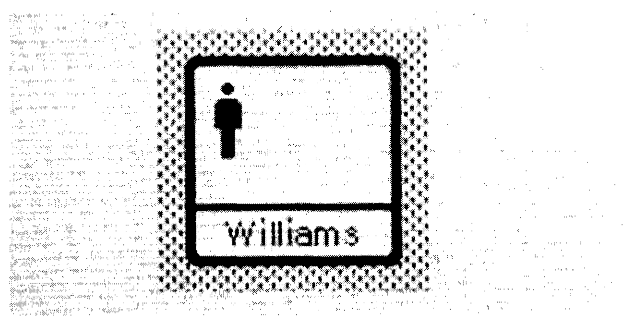

Figure 10—A user icon

use last names for user identification. Anyone named Smith, as is one of the authors, knows that this doesn't work. When he first became a user on such a system, *Smith* had long ago been taken. In fact, "D. Smith" and even "D. C. Smith" had been taken. He finally settled on "DaveSmith", all one word, with which he has been stuck to this day. Needless to say, that is *not* how he identifies himself to people. In the future, people will not tolerate this kind of antihumanism from computers. Star already does better: it follows society's conventions. User icons provide unambiguous unique references to individual people, using their normal names. The information about users, and indeed about all network resources, is physically stored in the Clearinghouse, a distributed database of names. In addition to a person's name in the ordinary sense, this information includes the name of the organization (e.g., Xerox, General Motors) and the name of the user's division within the organization. A person's linear name need be unique only within his division. It can be fully spelled out if necessary, including spaces and punctuation. Aliases can be defined. User icons are *references* to this information. You need not even know, let alone type, the unique linear representation for a user; you need only have the icon.

User group

User group icons (Figure 11) contain individual users and/or other user groups. They allow you to organize people according to various criteria. User groups serve both to control

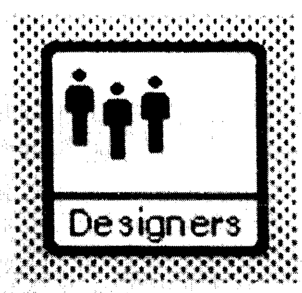

Figure 11—A user group icon

access to information such as file drawers (access control lists) and to make it easy to send mail to a large number of people (distribution lists). The latter is becoming increasingly important as more and more people start to take advantage of computer-assisted communication. At Xerox we have found that as soon as there were more than a thousand Alto users, there were almost always enough people interested in any topic whatsoever to form a distribution list for it. These user groups have broken the bonds of geographical proximity that have historically limited group membership and communication. They have begun to turn Xerox into a nationwide "village," just as the Arpanet has brought computer science researchers around the world closer together. This may be the most profound impact that computers have on society.

Calculator

A variety of styles of calculators (Figure 12) let you perform arithmetic calculations. Numbers can be moved between Star documents and calculators, thereby reducing the amount of typing and the possibility of errors. Rows or columns of tables can be summed. The calculators are user-tailorable and extensible. Most are modeled after pocket calculators—business, scientific, four-function—but one is a tabular calculator similar to the popular Visicalc program.

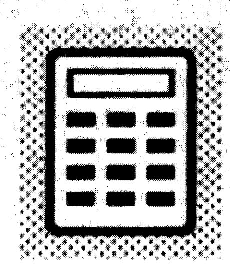

Figure 12—A calculator icon

Terminal emulators

The terminal emulators permit you to communicate with existing mainframe computers using existing protocols. Initially, teletype and 3270 terminals are emulated, with additional ones later (Figure 13). You open one of the terminal icons and type into its window; the contents of the window behave exactly as if you were typing at the corresponding terminal. Text in the window can be copied to and from Star documents, which makes Star's rich environment available to them.

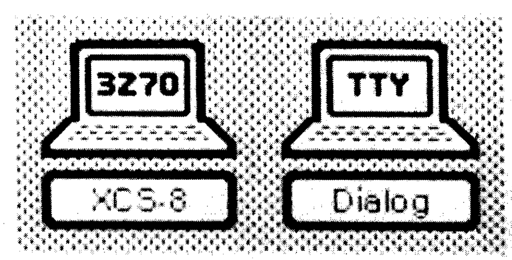

Figure 13—3270 and TTY emulation icons

Directory

The Directory provides access to network resources. It serves as the source for icons representing those resources; the Directory contains one icon for each resource available (Figure 14). When you are first registered in a Star network,

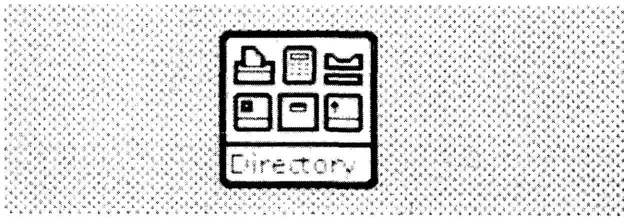

Figure 14—A Directory icon

your Desktop contains nothing but a Directory icon. From this initial state, you access resources such as file drawers, printers, and mail baskets by opening the Directory and copying out their icons. You can also get blank data icons out of the Directory. You can retrieve other data icons from file drawers. Star places no limits on the complexity of your Desktop except the limitation imposed by physical screen area (Figure 15). The Directory also contains Remote Directories representing resources available on other networks. These can be opened, recursively, and their resource icons copied out, just as with the local Directory. You deal with local and remote resources in exactly the same way.

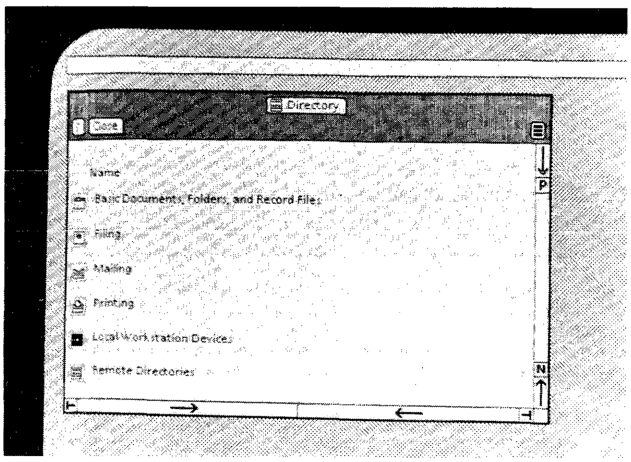

Figure 15—The Directory window, showing the categories of resources available

The important thing to observe is that although the functions performed by the various icons differ, the way you interact with them is the same. You select them with the mouse. You push the MOVE, COPY, or DELETE key. You push the OPEN key to see their contents, the PROPERTIES key to see their properties, and the SAME key to copy their properties. This is the result of rigorously applying the principle of uniformity to the design of icons. We have applied it to other areas of Star as well, as will be seen.

WINDOWS

Windows are rectangular areas that display the contents of icons on the screen. Much of the inspiration for Star's design came from Alan Kay's Flex machine[17] and his later Smalltalk programming environment on the Alto.[18] The Officetalk treatment of windows was also influential; in fact, Officetalk, an experimental office-forms-processing system on the Alto, provided ideas in a variety of areas.[19] Windows greatly increase the amount of information that can be manipulated on a display screen. Up to six windows at a time can be open in Star. Each window has a header containing the name of the icon and a menu of commands. The commands consist of a standard set present in all windows ("?", CLOSE, SET WINDOW) and others that depend on the type of icon. For example, the window for a record file contains commands tailored to information retrieval. CLOSE removes the window from the display screen, returning the icon to its tiny size. The "?" command displays the online documentation describing the type of window and its applications.

Each window has two scroll bars for scrolling the contents vertically and horizontally. The scroll bars have jump-to-end areas for quickly going to the top, bottom, left, or right end of the contents. The vertical scroll bar also has areas labeled N and P for quickly getting the next or previous screenful of the contents; in the case of a document window, they go to the next or previous page. Finally, the vertical scroll bar has a jumping area for going to a particular part of the contents, such as to a particular page in a document.

Unlike the windows in some Alto programs, Star windows do not overlap. This is a deliberate decision, based on our observation that many Alto users were spending an inordinate amount of time manipulating windows themselves rather than their contents. This manipulation of the medium is overhead, and we want to reduce it. Star automatically partitions the display space among the currently open windows. You can control on which side of the screen a window appears and its height.

PROPERTY SHEETS

At a finer grain, the Star world is organized in terms of *objects* that have *properties* and upon which *actions* are performed. A few examples of objects in Star are text characters, text paragraphs, graphic lines, graphic illustrations, mathematical summation signs, mathematical formulas, and icons. Every object has properties. Properties of text characters include type style, size, face, and posture (e.g., bold, italic). Properties of paragraphs include indentation, leading, and alignment. Properties of graphic lines include thickness and structure (e.g., solid, dashed, dotted). Properties of document icons include name, size, creator, and creation date. So the properties of an object depend on the type of the object. These ideas are similar to the notions of classes, objects, and messages in Simula[20] and Smalltalk. Among the editors that use these ideas are the experimental text editor Bravo[21] and the experimental graphics editor Draw,[22] both developed at the Xerox Palo Alto Research Center. These all supplied valuable knowledge and insight to Star. In fact, the text editor aspects of Star were derived from Bravo.

In order to make properties visible, we invented the notion of a property sheet (Figure 16). A property sheet is a two-dimensional formlike environment which shows the proper-

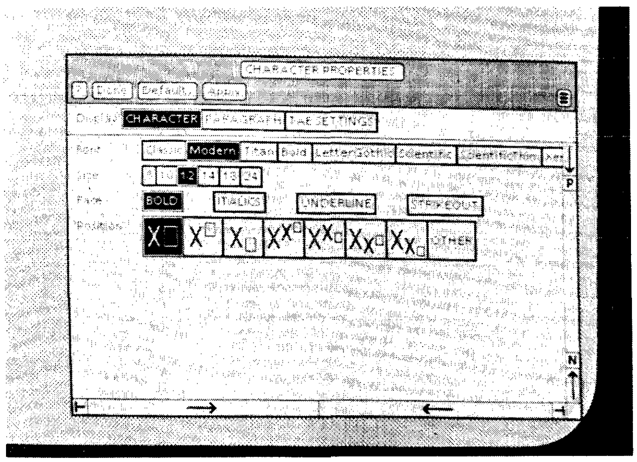

Figure 16—The property sheet for text characters

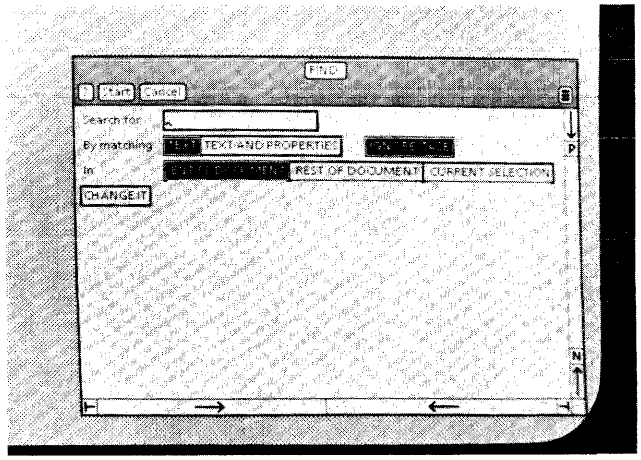

Figure 17—The option sheet for the Find command

ties of an object. To display one, you select the object of interest using the mouse and push the PROPERTIES key on the keyboard. Property sheets may contain three types of parameters:

1. *State*—State parameters display an independent property, which may be either on or off. You turn it on or off by pointing to it with the mouse and clicking a mouse button. When on, the parameter is shown video reversed. In general, any combination of state parameters in a property sheet can be on. If several state parameters are logically related, they are shown on the same line with space between them. (See "Face" in Figure 16.)
2. *Choice*—Choice parameters display a set of mutually exclusive values for a property. Exactly one value must be on at all times. As with state parameters, you turn on a choice by pointing to it with the mouse and clicking a mouse button. If you turn on a different value, the system turns off the previous one. Again the one that is on is shown video reversed. (See "Font" in Figure 16.) The motivation for state and choice parameters is the observation that it is generally easier to take a multiple-choice test than a fill-in-the-blanks one. When options are made visible, they become easier to understand, remember, and use.
3. *Text*—Text parameters display a box into which you can type a value. This provides a (largely) unconstrained choice space; you may type any value you please, within the limits of the system. The disadvantage of this is that the set of possible values is not visible; therefore Star uses text parameters only when that set is large. (See "Search for" in Figure 17.)

Property sheets have several important attributes:

1. A small number of parameters gives you a large number of combinations of properties. They permit a rich choice space without a lot of complexity. For example, the character property sheet alone provides for 8 fonts, from 1 to 6 sizes for each (an average of about 2), 4 faces (any combination of which can be on), and 8 positions relative to the baseline (including OTHER, which lets you type in a value). So in just four parameters, there are over $8 \times 2 \times 2^4 \times 8 = 2048$ combinations of character properties.
2. They show all of the properties of an object. None is hidden. You are constantly reminded what is available every time you display a property sheet.
3. They provide progressive disclosure. There are a large number of properties in the system as a whole, but you want to deal with only a small subset at any one time. Only the properties of the selected object are shown.
4. They provide a "bullet-proof" environment for altering the characteristics of an object. Since only the properties of the selected object are shown, you can't accidentally alter other objects. Since only valid choices are displayed, you can't specify illegal properties. This reduces errors.

Property sheets are an example of the Star design principle that *seeing and pointing* is preferred over *remembering and typing*. You don't have to remember what properties are available for an object; the property sheet will show them to you. This reduces the burden on your memory, which is particularly important in a functionally rich system. And most properties can be changed by a simple pointing action with the mouse.

The three types of parameters are also used in *option sheets*. (Figure 18). Option sheets are just like property sheets, except that they provide a visual interface for *arguments to commands* instead of *properties of objects*. For example, in the Find option sheet there is a text parameter for the string to search for, a choice parameter for the range over which to search, and a state parameter (CHANGE IT) controlling whether to replace that string with another one. When CHANGE IT is turned on, an additional set of parameters appears to contain the replacement text. This technique of having some parameters appear depending on the settings of others is another part of our strategy of progressive disclosure: hiding information (and therefore complexity) until it is needed, but making it visible when it is needed. The various sheets appear simpler than if all the options were always shown.

COMMANDS

Commands in Star take the form of noun-verb pairs. You specify the object of interest (the noun) and then invoke a command to manipulate it (the verb). Specifying an object is called *making a selection.* Star provides powerful selection mechanisms, which reduce the number and complexity of commands in the system. Typically, you exercise more dexterity and judgment in making a selection than in invoking a command. The ways to make a selection are as follows:

1. With the mouse—Place the cursor over the object on the screen you want to select and click the first (SELECT) mouse button. Additional objects can be selected by using the second (ADJUST) mouse button; it adjusts the selection to include more or fewer objects. Most selections are made in this way.
2. With the NEXT key on the keyboard—Push the NEXT key, and the system will select the contents of the next field in a document. Fields are one of the types of special higher-level objects that can be placed in documents. If the selection is currently in a table, NEXT will step through the rows and columns of the table, making it easy to fill in and modify them. If the selection is currently in a mathematical formula, NEXT will step through the various elements in the formula, making it easy to edit them. NEXT is like an intelligent step key; it moves the selection between semantically meaningful locations in a document.
3. With a command—Invoke the FIND command, and the system will select the next occurrence of the specified text, if there is one. Other commands that make a selection include OPEN (the first object in the opened window is selected) and CLOSE (the icon that was closed becomes selected). These optimize the use of the system.

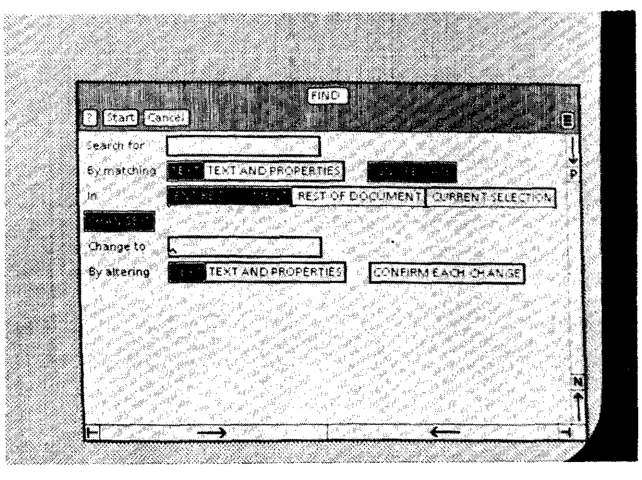

Figure 18—The Find option sheet showing Substitute options *(The extra options appear only when CHANGE IT is turned on)*

The object (noun) is almost always specified before the action (verb) to be performed. This makes the command interface *modeless;* you can change your mind as to which object to affect simply by changing the selection before invoking the command.[23] No "accept" function is needed to terminate or confirm commands, since invoking the command is the last step. Inserting text does not require a command; you simply make a selection and begin typing. The text is placed after the end of the selection. A few commands require more than one operand and hence are modal. For example, the MOVE and COPY commands require a destination as well as a source.

GENERIC COMMANDS

Star has a few commands that can be used throughout the system: MOVE, COPY, DELETE, SHOW PROPERTIES, COPY PROPERTIES, AGAIN, UNDO, and HELP. Each performs the same way regardless of the type of object selected. Thus we call them generic commands. For example, you follow the same set of actions to move text in a document as to move a document in a folder or a line in an illustration: select the object, move the MOVE key, and indicate the destination. Each generic command has a key devoted to it on the keyboard. (HELP and UNDO don't use a selection.)

These commands are more basic than the ones in other computer systems. They strip away extraneous application-specific semantics to get at the underlying principles. Star's generic commands are derived from *fundamental computer science concepts* because they also underlie operations in programming languages. For example, program manipulation of data structures involves moving or copying values from one data structure to another. Since Star's generic commands embody fundamental underlying concepts, they are widely applicable. Each command fills a host of needs. Few commands are required. This simplicity is desirable in itself, but it has another subtle advantage: it makes it easy for users to form a model of the system. What people can understand, they can use. Just as progress in science derives from simple, clear theories, so progress in the usability of computers depends on simple, clear user interfaces.

Move

MOVE is the most powerful command in the system. It is used during text editing to rearrange letters in a word, words in a sentence, sentences in a paragraph, and paragraphs in a document. It is used during graphics editing to move picture elements such as lines and rectangles around in an illustration. It is used during formula editing to move mathematical structures such as summations and integrals around in an equation. It replaces the conventional "store file" and "retrieve file" commands; you simply move an icon into or out of a file drawer or folder. It eliminates the "send mail" and "receive mail" commands; you move an icon to an Out basket or from an In basket. It replaces the "print" command; you move an icon to a printer. And so on. MOVE strips away much of the historical clutter of computer commands. It is more fundamental than the myriad of commands it replaces. It is simultaneously more powerful and simpler.

MOVE also reinforces Star's physical metaphor: a moved object can be in only one place at one time. Most computer file transfer programs only make copies; they leave the originals behind. Although this is an admirable attempt to keep information from accidentally getting lost, an unfortunate side effect is that sometimes you lose track of where the most recent information is, since there are multiple copies floating around. MOVE lets you model the way you manipulate information in the real world, should you wish to. We expect that during the *creation* of information, people will primarily use MOVE; during the *dissemination* of information, people will make extensive use of COPY.

Copy

COPY is just like MOVE, except that it leaves the original object behind untouched. Star elevates the concept of copying to the level of *a paradigm for creating*. In all the various domains of Star, you *create by copying*. Creating something out of nothing is a difficult task. Everyone has observed that it is easier to modify an existing document or program than to write it originally. Picasso once said, "The most awful thing for a painter is the white canvas.... To copy others is necessary."[24] Star makes a serious attempt to alleviate the problem of the "white canvas," to make copying a practical aid to creation. Consider:

- You create new documents by copying existing ones. Typically you set up blank documents with appropriate formatting properties (e.g., fonts, margins) and then use those documents as *form pad* sources for new documents. You select one, push COPY, and presto, you have a new document. The form pad documents need not be blank; they can contain text and graphics, along with fields for variable text such as for business forms.
- You place new network resource icons (e.g., printers, file drawers) on your Desktop by copying them out of the Directory. The icons are registered in the Directory by a system administrator working at a server. You simply copy them out; no other initialization is required.
- You create graphics by copying existing graphic images and modifying them. Star supplies an initial set of such images, called *transfer symbols*. Transfer symbols are based on the idea of dry-transfer rub-off symbols used by many secretaries and graphic artists. Unlike the physical transfer symbols, however, the computer versions can be modified: they can be moved, their sizes and proportions can be changed, and their appearance properties can be altered. Thus a single Star transfer symbol can produce a wide range of images. We will eventually supply a set of documents (transfer sheets) containing nothing but special images tailored to one application or another: people, buildings, vehicles, machinery. Having these as sources for graphics copying helps to alleviate the "white canvas" feeling.
- In a sense, you can even type characters by copying them from keyboard windows. Since there are many more characters (up to 2^{16}) in the Star character set than there are keys on the keyboard, Star provides a series of keyboard interpretation windows (Figure 19), which allow you to see and change the meanings of the keyboard keys. You are presented with the options; you look them over and choose the ones you want.

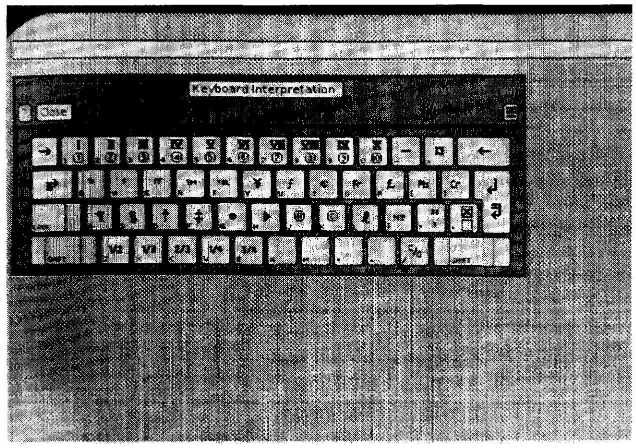

Figure 19—The Keyboard Interpretation window

This displays other characters that may be entered from the keyboard. The character set shown here contains a variety of common office symbols.

Delete

This deletes the selected object. If you delete something by mistake, UNDO will restore it.

Show Properties

SHOW PROPERTIES displays the properties of the selected object in a property sheet. You select the object(s) of interest, push the PROPERTIES (PROP'S) key, and the appropriate property sheet appears on the screen in such a position as to not overlie the selection, if possible. You may change as many properties as you wish, including none. When finished, you invoke the Done command in the property sheet menu. The property changes are applied to the selected objects, and the property sheet disappears. Notice that SHOW PROPERTIES is therefore used both to examine the current properties of an object and to change those properties.

Copy Properties

You need not use property sheets to alter properties if there is another object on the screen that already has the desired properties. You can select the object(s) to be changed, push the SAME key, then designate the object to use as the source. COPY PROPERTIES makes the selection look the "same" as the source. This is particularly useful in graphics editing. Frequently you will have a collection of lines and symbols whose appearance you want to be coordinated (all the same line width, shade of grey, etc.). You can select all the objects to be changed, push SAME, and select a line or symbol having the desired appearance. In fact, we find it helpful to set up a document with a variety of graphic objects in a variety of appearances to be used as sources for copying properties.

Again

AGAIN repeats the last command(s) on a new selection. All the commands done since the last time a selection was made are repeated. This is useful when a short sequence of commands needs to be done on several different selections; for example, make several scattered words bold and italic and in a larger font.

Undo

UNDO reverses the effects of the last command. It provides protection against mistakes, making the system more forgiving and user-friendly. Only a few commands cannot be repeated or undone.

Help

Our effort to make Star a personal, self-contained system goes beyond the hardware and software to the tools that Star provides to teach people how to use the system. Nearly all of its teaching and reference material is on line, stored on a file server. The Help facilities automatically retrieve the relevant material as you request it.

The HELP key on the keyboard is the primary entrance into this online information. You can push it at any time, and a window will appear on the screen displaying the Help table of contents (Figure 20). Three mechanisms make finding information easier: *context-dependent invocation, help references,* and a *keyword search command.* Together they make the online documentation more powerful and useful than printed documentation.

- *Context-dependent invocation*—The command menu in every window and property/option sheet contains a "?" command. Invoking it takes you to a part of the Help documentation describing the window, its commands, and its functions. The "?" command also appears in the message area at the top of the screen; invoking that one takes you to a description of the message (if any) currently in the message area. That provides more detailed explanations of system messages.
- *Help references*—These are like menu commands whose effect is to take you to a different part of the Help material. You invoke one by pointing to it with the mouse, just as you invoke a menu command. The writers of the material use the references to organize it into a network of interconnections, in a way similar to that suggested by Vannevar Bush[25] and pioneered by Doug Engelbart in his NLS system.[26,27] The interconnections permit cross-referencing without duplication.
- The *SEARCH FOR KEYWORD command*—This command in the Help window menu lets you search the available documentation for information on a specific topic. The keywords are predefined by the writers of the Help material.

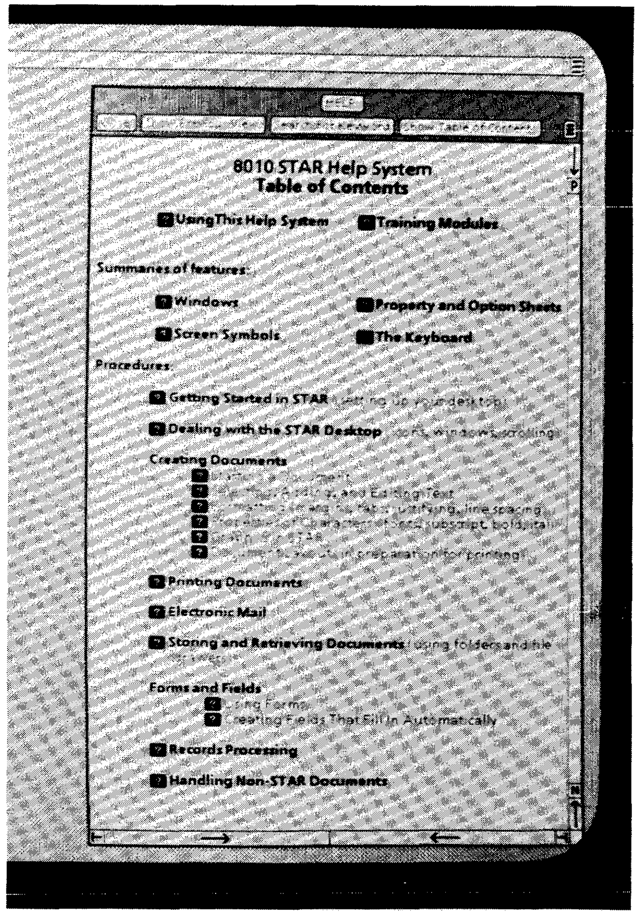

Figure 20—The Help window, showing the table of contents

Selecting a square with a question mark in it takes you to the associated part of the Help documentation.

SUMMARY

We have learned from Star the importance of formulating the user's conceptual model *first, before* software is written, rather than tacking on a user interface *afterward.* Doing good user interface design is not easy. Xerox devoted about thirty work-years to the design of the Star user interface. It was designed *before* the functionality of the system was fully decided. It was designed *before* the computer hardware was even built. We worked for two years *before* we wrote a single line of actual product software. Jonathan Seybold put it this way: "Most system design efforts start with hardware specifications, follow this with a set of functional specifications for the software, then try to figure out a logical user interface and command structure. The Star project started the other way around: the paramount concern was to define a conceptual model of how the user would relate to the system. Hardware and software followed from this."[4]

Alto served as a valuable prototype for Star. Over a thousand Altos were eventually built, and Alto users have had several thousand work-years of experience with them over a period of eight years, making Alto perhaps the largest prototyping effort in history. There were dozens of experimental programs written for the Alto by members of the Xerox Palo Alto Research Center. Without the creative ideas of the authors of those systems, Star in its present form would have been impossible. On the other hand, it was a real challenge to bring some order to the different user interfaces on the Alto. In addition, we ourselves programmed various aspects of the Star design on Alto, but every bit (sic) of it was throwaway code. Alto, with its bit-mapped display screen, was powerful enough to implement and test our ideas on visual interaction.

REFERENCES

1. Smith, D. C., E. F. Harslem, C. H. Irby, R. B. Kimball, and W. L. Verplank. "Designing the Star User Interface." *Byte,* April 1982.
2. Metcalfe, R. M., and D. R. Boggs. "Ethernet: Distributed Packet Switching for Local Computer Networks." *Communications of the ACM,* 19 (1976), pp. 395–404.
3. Intel, Digital Equipment, and Xerox Corporations. "The Ethernet, A Local Area Network: Data Link Layer and Physical Layer Specifications (version 1.0)." Palo Alto: Xerox Office Products Division, 1980.
4. Seybold, J. W. "Xerox's 'Star.'" *The Seybold Report.* Media, Pennsylvania: Seybold Publications, 10 (1981), 16.
5. Thacker, C. P., E. M. McCreight, B. W. Lampson, R. F. Sproull, and D. R. Boggs. "Alto: A Personal Computer." In D. Siewiorek, C. G. Bell, and A. Newell (eds.), *Computer Structures: Principles and Examples.* New York: McGraw-Hill, 1982.
6. Ingalls, D. H. "The Smalltalk Graphics Kernel." *Byte,* 6 (1981), pp. 168–194.
7. English, W. K., D. C. Engelbart, and M. L. Berman. "Display-Selection Techniques for Text Manipulation." *IEEE Transactions on Human Factors in Electronics,* HFE-8 (1967), pp. 21–31.
8. Fitts, P. M. "The Information Capacity of the Human Motor System in Controlling Amplitude of Movement." *Journal of Experimental Psychology,* 47 (1954), pp. 381–391.
9. Card, S., W. K. English, and B. Burr. "Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys, and Text Keys for Text Selection on a CRT." *Ergonomics,* 21 (1978), pp. 601–613.
10. Oppen, D. C., and Y. K. Dalal. "The Clearinghouse: A Decentralized Agent for Locating Named Objects in a Distributed Environment." Palo Alto: Xerox Office Products Division, OPD-T8103, 1981.
11. Huggins, W. H., and D. Entwisle. *Iconic Communication.* Baltimore and London: The Johns Hopkins University Press, 1974.
12. Smith, D. C. *Pygmalion, A Computer Program to Model and Stimulate Creative Thought.* Basel and Stuttgart: Birkhäuser Verlag, 1977.
13. Bolt, R. *Spatial Data-Management.* Cambridge, Massachusetts: Massachusetts Institute of Technology Architecture Machine Group, 1979.
14. Sutherland, I. "Sketchpad, A Man-Machine Graphical Communication System." *AFIPS, Proceedings of the Fall Joint Computer Conference* (Vol. 23), 1963, pp. 329–346.
15. Sutherland, W. "On-Line Graphical Specifications of Computer Procedures." Cambridge, Massachusetts: Massachusetts Institute of Technology, 1966.
16. Christensen, C. "An Example of the Manipulation of Directed Graphs in the AMBIT/G Programming Language." In M. Klerer and J. Reinfelds (eds.), *Interactive Systems for Experimental and Applied Mathematics.* New York: Academic Press, 1968.
17. Kay, A. C. *The Reactive Engine.* Salt Lake City: University of Utah, 1969.
18. Kay, A. C., and the Learning Research Group. "Personal Dynamic Media." Xerox Palo Alto Research Center Technical Report SSL-76-1, 1976. (A condensed version is in *IEEE Computer,* March 1977, pp. 31–41.)
19. Newman, W. M. "Officetalk-Zero: A User's Manual." Xerox Palo Alto Research Center Internal Report, 1977.
20. Dahl, O. J., and K. Nygaard. "SIMULA–An Algol-Based Simulation Language." *Communications of the ACM,* 9 (1966), pp. 671–678.
21. Lampson, B. "Bravo Manual." In *Alto User's Handbook,* Xerox Palo Alto Research Center, 1976 and 1978. (Much of the design and all of the implementation of Bravo was done by Charles Simonyi and the skilled programmers in his "software factory.")
22. Baudelaire, P., and M. Stone. "Techniques for Interactive Raster Graphics." *Proceedings of the 1980 Siggraph Conference,* 14 (1980), 3.
23. Tesler, L. "The Smalltalk Environment." *Byte,* 6 (1981), pp. 90–147.
24. Wertenbaker, L. *The World of Picasso.* New York: Time-Life Books, 1967.
25. Bush, V. "As We May Think." *Atlantic Monthly,* July 1945.
26. Engelbart, D. C. "Augmenting Human Intellect: A Conceptual Framework." Technical Report AFOSR-3223, SRI International, Menlo Park, Calif., 1962.
27. Engelbart, D. C., and W. K. English. "A Research Center for Augmenting Human Intellect." *AFIPS Proceedings of the Fall Joint Computer Conference* (Vol. 33), 1968, pp. 395–410.