The integrated software and user interface of Apple's Lisa

by EDWARD W. BIRSS
Apple Computer, Inc.
Cupertino, California

ABSTRACT

In 1979 Apple began to develop Lisa, a workstation to enhance the productivity of office workers. The hardware was built around a Motorola 68000, a bit-mapped display, and a mouse. The user interface is intuitive, using real-world concepts rather than computer concepts. It is easy to learn, and provides for both novice users still learning the system and users that have mastered the system. The user interface is modeless and consistent. The uniformity of the user interface supports transferrable learning—the ability to learn an operation once and apply it over and over again in another application in a different context.

The user interface also supports data interchange among documents of the same or different types. This interchange of data, coupled with the multitasking operating system and the multiple windows of the Lisa, permits the use of several tools to perform a task that one tool alone could not accomplish. The Lisa user interface and its applications provide an environment that allows the user to concentrate on what is to be accomplished rather than on how to accomplish it. In this way, Lisa provides tools to improve the productivity of the office worker.

INTRODUCTION

Apple Computer formed the Lisa team in 1979 to develop a personal computer that would dramatically improve the productivity of typical office workers (professionals, managers, and their assistants). To accomplish this goal, a hardware and software solution radically different from current personal computer offerings was required. At that time, personal computers had the functionality but lacked the capacity, speed, and ease of use necessary to reach a market of users who did not want to learn the details of how a computer worked.

Inspired by SMALLTALK[1] the Lisa team developed a system that has the functionality and speed users require, and additionally has a common user interface that supports gradual learning and promotes interchange of data among the same or different applications. The combination of multiple tools with a consistent user interface and data interchange among applications permits the user to work with several tools concurrently to accomplish a particular task.

LISA HARDWARE

The Lisa is a Motorola 68000-based personal computer with 512 or 1024 Kbytes of main memory, a memory management unit, a bit-mapped display, a detachable keyboard, a mouse, a built-in 400-Kbyte floppy disk drive, and a 5- or 10-megabyte Winchester disk (see Figure 1). This hardware provides the functionality, speed, and ease of use required to support the Lisa user interface.

The 68000 microprocessor was not the first choice. Development began on a home-grown bit-sliced system to provide the computing power. When the 68000 became available in sample quantities, we evaluated it and found it had good performance and was more economical.

The memory management unit (MMU) provides different logical address space contexts for processes and protection. The protection ensures that an individual application fault does not damage the rest of the system and therefore improves system reliability. The MMU also provides for code segment faulting and automatic stack expansion.

The bit-mapped display provides graphics and text support needed for the user interface. The display is 720 by 364 pixels and supports quality graphics and text fonts of different sizes and faces. This permits the word processing applications to use black on white images, proportional-spaced fonts, and different type styles including boldface, underline, and italic.

To complement the graphics output, Apple wanted to use a mouse for a graphics input device, but existing ones were unreliable and had precision bearings that made them expensive and difficult to manufacture. Apple developed a mouse that is precise, tracks on almost any surface, and is easy to manufacture. The original prototypes had three buttons, but we found users spent too much time looking away from the screen to determine which button to push; consequently we changed to a two-button mouse. Once we found alternative ways to implement the functions of the second button, we changed to a one-button mouse.

LISA SOFTWARE

Lisa's Desktop Model

The office system software provides the user with a desktop that mirrors the function of a desk in the office. On the Lisa desktop, icons (small pictures) depict the office world. In Figure 2 we see a variety of icons, a menu bar at the very top of the screen, and two windows. One of the windows contains the contents of a LisaWrite document, and the other contains the catalogue of a disk.

There are several types of icons: documents, stationery pads, folders, a wastebasket, a ProFile disk, a clipboard, and one called Preferences. The types of documents are spreadsheets, business charts, lists, text documents, etc. These document icons quickly show the user not only that the object represented is a document, but also what type of document. Stationery pads permit a user to create new documents, and in addition permit a user to configure predefined forms or templates. For example, an office usually has different types of paper stock. One might be used for letters going outside the office, and another for interoffice memoranda. In the Lisa model, a user sets up a stationery pad for both types, and then each time the user needs to write a letter, he simply tears off a new piece of letter stock from the appropriate stationery pad. Since the pad is constructed from a document, the stock can be set up with the desired initial format and content. Folders provide a convenient way of grouping logically related documents together—similar to the function of file folders in the office. Thus, folders organize the contents of diskettes, disks, and the desktop.

The desktop supports two icons that represent storage devices. The ProFile icon represents the Winchester disk drive, and the diskette icon (not shown) represents a floppy diskette inserted in the built-in drive. These devices are used for document and program storage.

The wastebasket is used to throw away documents and programs. Just as the office worker can retrieve something thrown away in the wastebasket, the Lisa user can retrieve objects thrown in the wastebasket.

The clipboard is used by the editing operations. When a user edits a document, pieces of information are placed on the

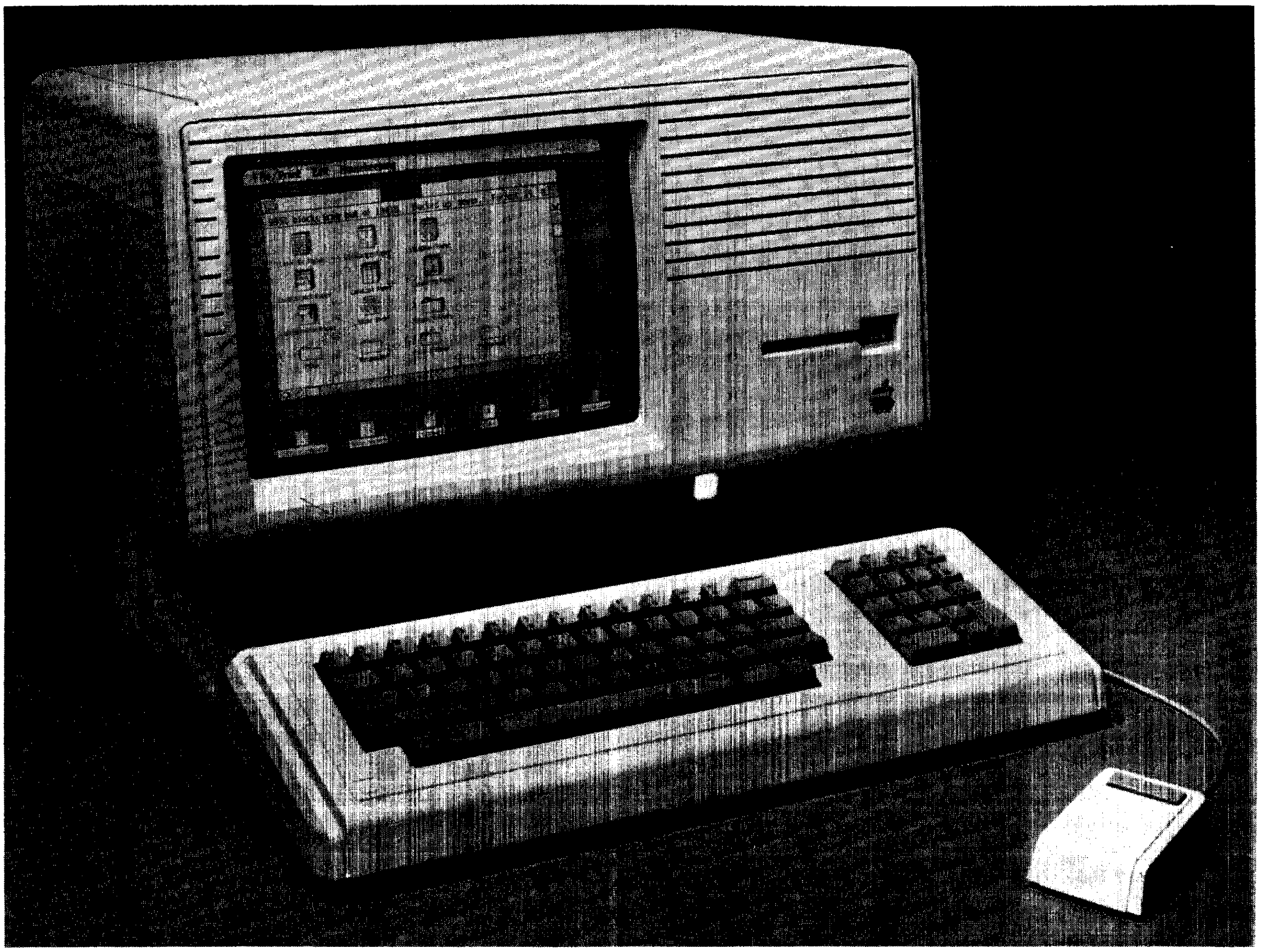


Figure 1—The Lisa

clipboard. This information can be copied into a different place in that document or into a different document altogether. Thus the clipboard acts as temporary storage for these scraps of information (more on this later).

The Preferences tool permits the user to customize the Lisa to suit his tastes. The user can set the screen brightness, the tone generator volume, the key repeat rate, the mouse click delay time, etc. Using Preferences, the user also can configure printers and disks.

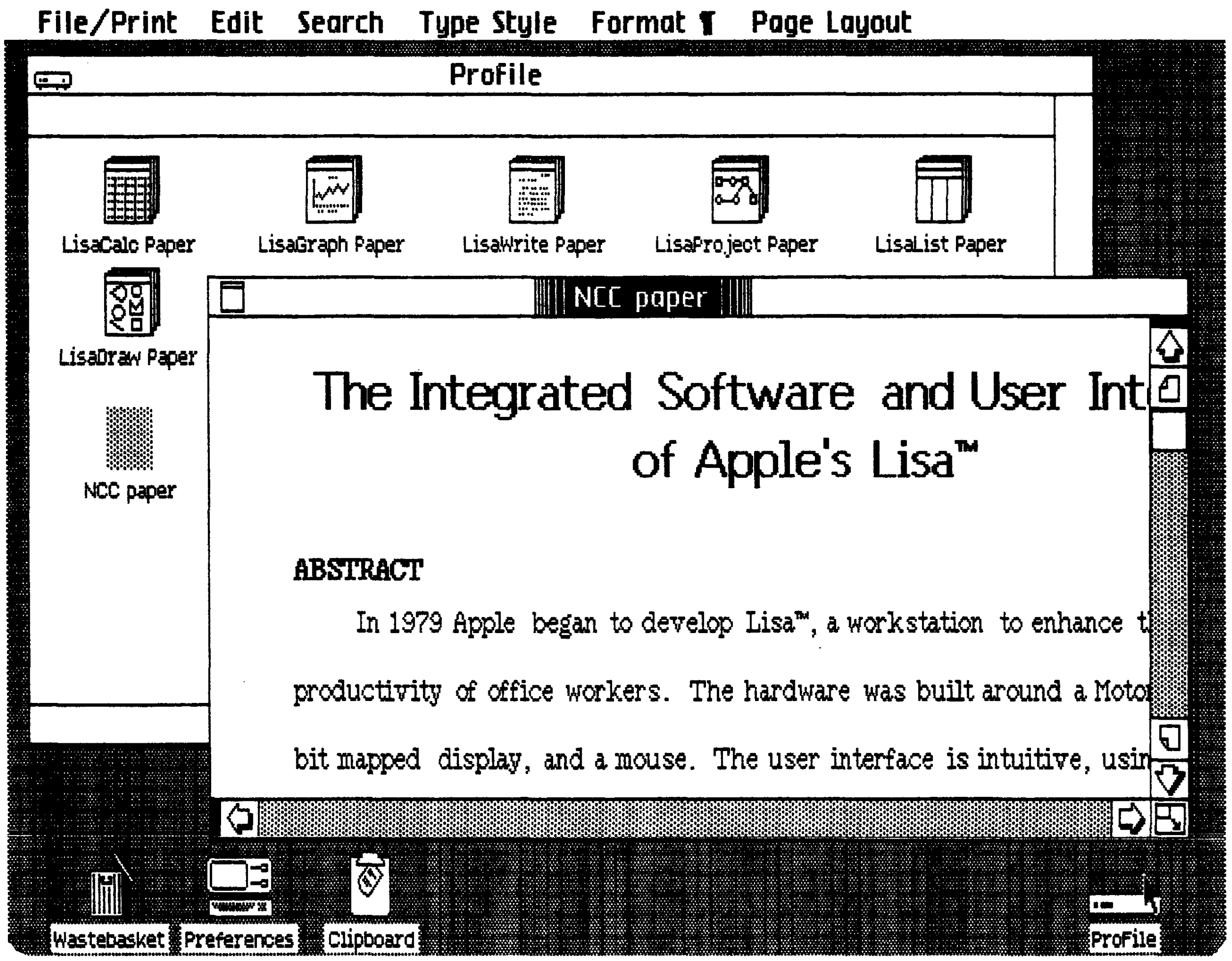
The menu bar, located at the top of the screen, shows the titles of the available pull-down menus. The menus are called pull-down because when a user depresses the mouse button over a menu title, a rectangular area under the menu title pulls down like a roller blind. The rectangular area is called a menu and contains a number of labels, which are called menu items. The user moves the mouse down through the menu items and selects the desired operation. The menus, in conjunction with the current selection, give the user the ability to specify actions. For example, one changes a word in a document to italic type by selecting the word and then choosing the italic item from the Type Style pull-down menu.

The example desktop also shows two windows. Windows in Lisa show the contents of disks, documents, wastebasket, etc. Lisa displays up to 20 windows at a time, and windows can overlap or completely obscure other windows. The user has full control over the size and position of the windows.

The User Interface Philosophy

The Lisa user interface is much more than just a mouse, bit-map graphics, a desktop with icons, and overlapping windows. The Lisa user interface is designed to be intuitive. It uses real-world concepts, not computer concepts, and provides familiar office objects and ideas. The natural model enables a user to try things out that would make sense in the real world. In general they directly transfer to Lisa's desktop world.

The user interface is designed to work the way you would expect it to work. In the office, users open documents, move them around, edit them, file them, etc. With Lisa, the mouse is used to manipulate objects directly. This is one of the key features of the Lisa user interface and is in stark contrast to traditional "computerese" of command languages and tex-



The Integrated Software and User Int
of Apple's Lisa™

ABSTRACT

In 1979 Apple began to develop Lisa™, a workstation to enhance t
productivity of office workers. The hardware was built around a Moto
bit mapped display, and a mouse. The user interface is intuitive, usin

Figure 2—A Lisa screen showing the menu bar, two windows, and several icons

tual, mode-driven menus. Because there is no command language, very little typing is required to perform operations.

To ensure that Lisa is easy to use and learn, Apple developed LisaGuide, an interactive guide that teaches novices how to use the mouse as well as the basic principles of selection and menus. Once they have been through LisaGuide, they pick an application and start learning through actual use. This seems to be fairly successful; very few users will actually consult the manual.

The use of a common and consistent user interface provides for transferrable learning. The user interacts with the desktop and all applications in the same way. For example, titles of documents on the desktop are edited the same way as text within memos or numbers in LisaCalc. In addition to the editing model, the filing and printing models are the same across all applications. The time a user invests in learning the editing, filing, and printing operations immediately transfers over to the next application. Consequently, the second application is easier to learn than the first.

One of the features of the user interface is that it addresses those users learning the system and those that have mastered it. The novice can learn a few operations, just enough to accomplish his task. As the user becomes more proficient with the system, he can graduate to the more advanced uses of Lisa including shortcuts to make his interactions even more effective. In contrast to other systems, Lisa does not burden the expert user with features intended for beginners.

Using Lisa

Wherever possible in Lisa, the user moves the mouse to manipulate objects directly. For example, to move a document from one diskette to another, the user moves the mouse over the icon representing the document, depresses the mouse button, moves the mouse (and the document icon) over the appropriate container, and releases the mouse button. The mouse moves windows around, sizes them, scrolls the contents of a window, and uses the elevator to jump to a position in the document. The elevator is a rectangular white icon in the scroll bar found at the bottom and right of the active window.

Another aspect of direct operation is the modeless nature of the user interface. A modeless system is a flat, nonhierarchical model, permitting virtually any operation at any time. Thus the user need not remember what mode to enter

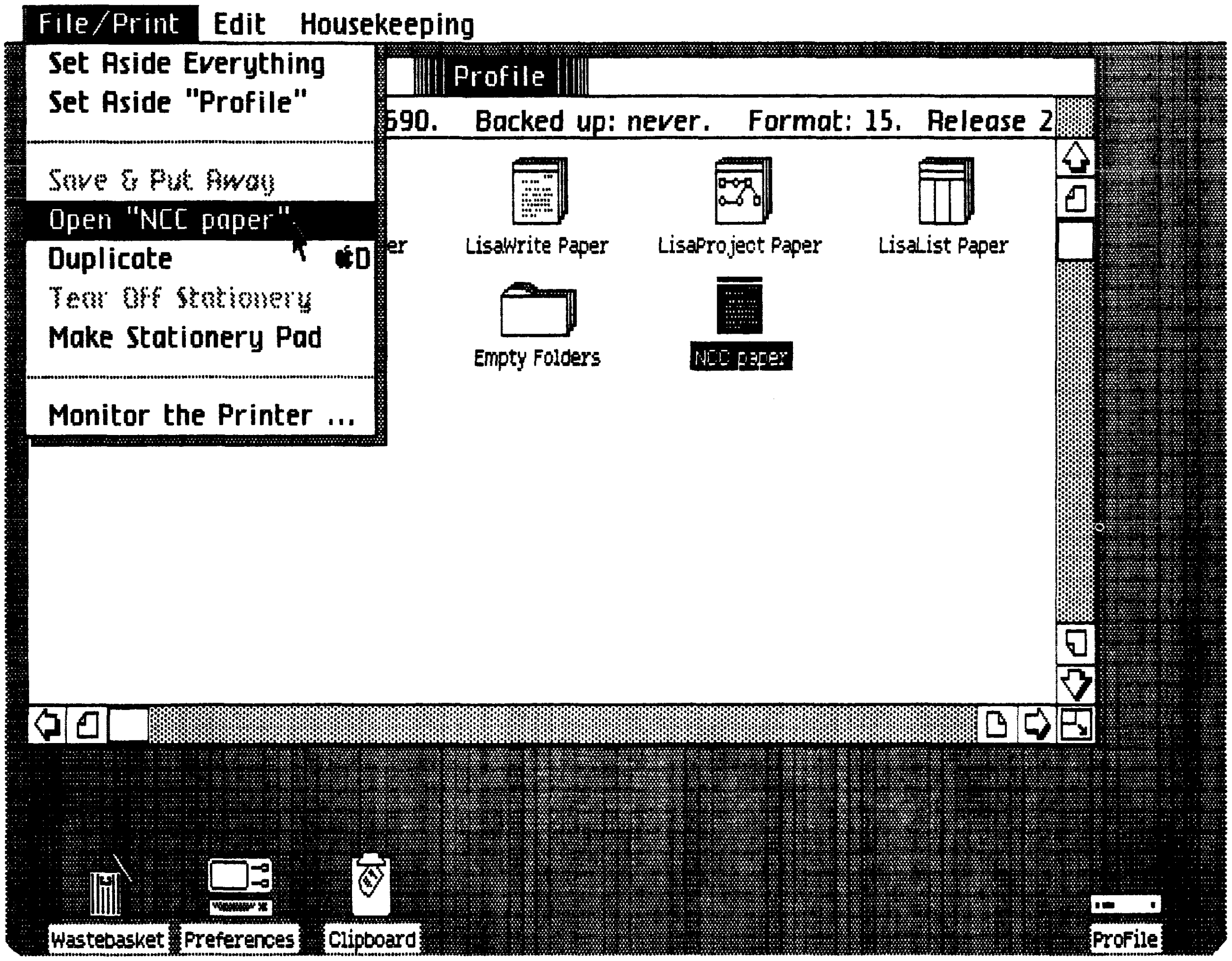


Figure 3—A Lisa screen showing a pull-down menu

to perform a function, nor what command to use to exit the mode. This flat command structure also permits the user to peruse the menus to find the most suitable operation.

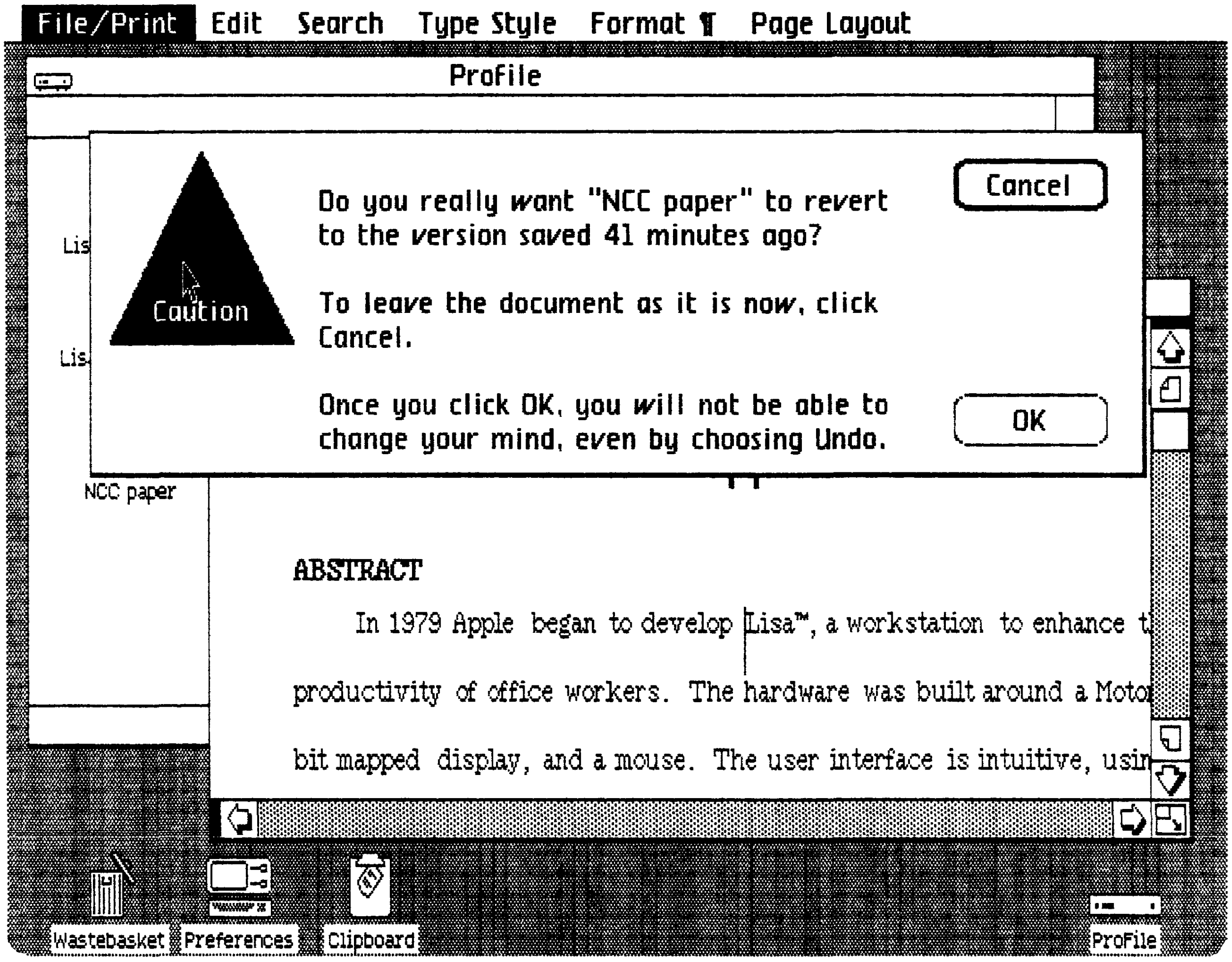
When the user wants to operate on objects in the Lisa environment, the model we use is to select the objects and then operate on them with an action selected from a menu. We call this the noun–verb model, and it permeates all the applications as well as the Desktop Manager. For example, to open a document, the user moves the mouse over the icon of the document (named NCC paper in Figure 3), and clicks the button once to select the document. Then the user moves the mouse up to the menu bar and depresses the mouse button over File/Print, which causes the menu to pull down (see Figure 3). The user moves the mouse (with the mouse button down) over the "Open NCC paper" menu item, and then releases the mouse button.

In addition to a single click to select an object, several objects can be selected with single-click drag. This operation proceeds as follows: First the user positions the mouse to one side of the object, then the user depresses the mouse button and moves (drags) the mouse through the objects. When the selection includes all the objects, the user releases the mouse button. The selection of objects with single click and single-click drag, combined with menu commands can be used to perform every operation.

Two types of shortcuts are provided for the expert user—multiple clicks of the mouse button (in quick succession) and Apple keys. Double- and triple-click operations substitute for selecting an object and operating on the object with a specific frequently used command. For example, a double click on a document icon opens the document. For less frequently used operations some menu commands have Apple key sequences. For example, text-editing menu commands like cut, copy, and paste have Apple key sequences that cause the command to be invoked. Apple key sequences involve holding down the Apple key along with an alphabetic character. Thus, multiple click shortcuts are used for the most frequently used operations on the selection, and the Apple key shortcuts are used for frequently used menu commands.

Error Handling

A good user interface has good error handling. There are three aspects of error handling in Lisa: error prevention, error notification, and error recovery.



Do you really want "NCC paper" to revert to the version saved 41 minutes ago?

To leave the document as it is now, click Cancel.

Once you click OK, you will not be able to change your mind, even by choosing Undo.

ABSTRACT

In 1979 Apple began to develop Lisa™, a workstation to enhance t

productivity of office workers. The hardware was built around a Moto

bit mapped display, and a mouse. The user interface is intuitive, usin

Figure 4—A Lisa screen showing an alert

With many personal computers the contents of diskettes can be damaged by ejecting a diskette or turning off the machine at an inopportune moment. To protect against such errors, Lisa has software-controlled mechanisms for the diskette eject and on–off button. To eject a diskette, the user selects the eject menu item, the software then suspends the processing of all the documents that reside on that diskette and writes out those suspended documents to the diskette. Once all the I/O has been completed, the diskette is ejected. Pushing the on–off button causes suspension of all documents, the ejection of all diskettes, and finally the power down of the Lisa. These controls help to ensure the integrity of the user's data.

Error notification in Lisa is handled with a special window called an alert (see Figure 4). An alert appears whenever the user must be notified that an error has occurred, a requested operation cannot be performed, or an explanation must be given. There are several kinds of alerts: stop, caution, note, ask, and wait. Stop alerts are used when the requested operation cannot be performed. Caution alerts inform the user that an operation has ramifications, and gives the user the opportunity to change his mind. Note alerts notify the user of something, ask alerts solicit input from the user, and wait alerts tell the user to wait until a lengthy operation completes. Alert messages that inform the user that an error has occurred have three parts. First, the user is told the nature of the problem; second, the user is told how to work around the problem; and finally, the user is told where to refer in the manual for more help.

Another level of recovery is provided by the Revert to Previous Version menu command. Sometimes a user makes several changes to a document and then changes his mind. In this event, the user can invoke the Revert to Previous Version command to return the document to the state when the document was last saved.

In the event of a program failure in Lisa, an alert message appears informing the user that the tool failed. The user is given the option to redisplay the document and if he chooses to do so, the document is shown, usually with the last changes intact.

The final area of error recovery is recovery from external errors such as power failures. If power goes off while using a computer, the disk is likely to be inconsistent; some of the current data are in memory but not on the disk. To protect against failures of this kind, the Lisa file system has redundant information permitting reconstruction of files on disk. The

Desktop Manager and the LisaList tool also detect such failures and repair and reconstruct their information. When the user powers up the Lisa after such a failure or opens a damaged LisaList document, he is informed that repair is needed. When the user confirms that the repair operation should be started, the repair begins.

Lisa Applications

Apple offers seven Lisa applications (also called tools): LisaCalc (spreadsheet), LisaWrite (word processing), LisaGraph (business graphics), LisaDraw (graphics editing), LisaProject (project scheduling), LisaList (list management), and LisaTerminal (terminal emulation). Lisa is an open system that permits third parties to develop Lisa applications, hence additional Lisa applications also are available.

The Lisa tools have effective user interfaces for their applications. LisaCalc and LisaGraph use the spreadsheet user interface developed by VisiCalc, further improved by the addition of the mouse. LisaList builds upon the user interface techniques developed in QBE.[2] LisaWrite uses techniques found in several word processors.

The unique Lisa applications are LisaDraw and LisaProject. LisaDraw is a structured graphics editor that permits the user to draw lines, circles, ovals, rectangles (both square and rounded), polygons, freehand curves, and text. It is used for such diverse applications as preparing diagrams for presentations and architectural drawings.

LisaProject is a PERT/CPM project-scheduling tool. It uses a graphical PERT chart representation to enter a project schedule. The user draws the schedule using rectangles for tasks and circles for milestones. The user specifies the task, the resources needed to accomplish it, the duration of the task, and its relation to other tasks. As tasks are added, durations changed, or scheduled dates specified, the schedule is recalculated.

The user interfaces of LisaDraw and LisaProject have opened up these tools to a much greater audience. Just as VisiCalc and QBE opened up spreadsheets and databases to those who were unable to use other offerings, LisaDraw and LisaProject have done the same in their application areas. Both these tools magnify the capabilities of the user. For example, people like myself who are totally inept at drawing are assisted by LisaDraw to the extent that very respectable results are easy to achieve. A similar result occurs with LisaProject. Administrative assistants unable to use conventional project-scheduling tools are now using LisaProject to make very large schedules.

During the development of the Lisa applications and the application libraries, we found that application development was not as easy as we would like it to be. The library structure was very hierarchical and was hard to use. Consequently we were determined to make it easier for third parties to develop Lisa applications. This led to two ways to develop Lisa applications, QuickPort and ToolKit.

QuickPort permits a third-party software developer to run standard PASCAL programs (ones that use standard PASCAL I/O) in a window in the Lisa office system. In addition, QuickPort permits cutting, copying, and pasting of information from the QuickPort window to other Lisa desktop windows. The modifications the third-party developer must make to the application to use QuickPort are minimal. The developer must use a few new units, and possibly make name conflict changes. This process can be accomplished in an afternoon.

QuickPort is the easiest way to get an application operational in the office system, but such a program cannot use all of the capabilities of the Lisa. ToolKit is used to write an application that fully uses the features of the Lisa. ToolKit is essentially a generic application that calls application-specific code to implement application-specific functions. This permits the sharing of common control structure code across several different ToolKit applications.

Because different applications have different needs, the ToolKit generic application had to be extremely flexible. The flexibility required, along with the need to call application-specific code, led to the use of classes similar to those in SMALLTALK.[1] The classes provide the ability to call the application-specific code while also permitting the developer to override or subclass a class to modify its behavior.

Both QuickPort and ToolKit promote the development of Lisa applications. This open nature of the Lisa office system permits third-party developers to develop a specific application, yet leverage off other Lisa applications. These third parties can develop applications that target specific markets while relying on the standard tools such as LisaWrite and LisaDraw for presentation of the results.

Integration in Lisa

Several components of integration in the Lisa system have already been mentioned. There is a consistent user interface that is common across all applications and the Desktop Manager. If the user wants to enter text and makes a mistake entering it, he can fix it using the standard text-editing model. The user does not have to remember which tool he is in, nor does he have to run an editor tool.

The editing model used by Lisa is the cut-and-paste model. Just as an editor might cut up a paragraph with scissors and paste-up sentences or paragraphs to improve an article, the Lisa user can cut and paste with the Clipboard. The editing model also includes the ability to copy to the Clipboard and undo the last edit operation. When a user copies or cuts an object, the object is copied onto the clipboard, a repository for scraps of information. The Paste command pastes the information from the clipboard into the active window replacing the current selection. This model is used for all objects; e.g., text within textual documents, numbers and formulas within LisaCalc, and graphics within LisaDraw. A user can cut or copy information from a paragraph and paste it into the same document or a different document. This copy and paste model is also the mechanism for data interchange between documents.

This data interchange is illustrated by the following example. Let us assume that a user has data in a LisaGraph document (a bar chart) and wishes to move them to a LisaDraw document to annotate and customize the chart for a presentation. To do this the user selects the entire graph in the Lisa-

Graph document, and then uses the Copy command in the Edit menu to copy the graph to the Clipboard. Next the user opens the LisaDraw document and selects Paste from the Edit menu. The user can then use the capabilities of the LisaDraw tool to add labels, change patterns, etc.

The direction of software integration that we see for Lisa in the future is presented in the following scenario. Lisa provides an environment where one can use LisaTerminal and gather data from a mainframe, copy the data to LisaList and subset them, copy the result to LisaCalc and perform some arithmetic manipulation to analyze the data, copy the resulting data to LisaGraph to make a chart, then copy the chart to LisaDraw to further customize it, and finally copy that customized chart to LisaWrite for inclusion in a report.

This type of integration permits the user to choose the best tools for his task. The user is free to concentrate on his task, not on the mechanics of typing the data or mastering the commands of the application. The model permits the tool's developer to concentrate more on the functionality that has to be provided, and not on extraneous features. For example, the LisaGraph tool can concentrate on drawing the best pie charts, without having to provide all the presentation flexibility (e.g., detached pie segments), since the chart can be copied to LisaDraw where the chart can be customized.

The open nature of the Lisa office system further expands the integration possibilities. The power of an open system becomes apparent when third-party tools can be used as equals in conjunction with the standard tools. This permits the Lisa to be used by a broader range of customers for a wider variety of applications. This is in contrast to closed systems, which are typically one large program, and do not permit integration of other components.

The Clipboard

The clipboard provides a common interchange form between documents. There are three distinct interchange forms—text, tables, and graphics. In most cases, the user is unaware of what type of interchange form is employed. The user merely selects the object, and copies or cuts it. The type of object determines the form of the data on the clipboard. In cases where a particular selection is ambiguous, the user is required to further specify his intent.

The cut, copy, and paste model does require that it be easy to move data from one tool's document to another. In the Lisa environment, this is provided by the multitasking operating system.[4] Lisa uses a multitasking operating system to permit the concurrent operation of processes. For example, this permits LisaGraph and LisaDraw documents to be both on the screen and in memory. When the user switches from one document to the other, the operation can complete rapidly (one or two seconds if both are in memory already).

The clipboard provides for rapid exchange of data to and from the same or different documents. It also provides the ability to undo the last cut or copy. In this way the user can undo a cut operation and then paste the previous clipboard contents. The astute reader will realize that only the last operation is undoable; undo of an undo undoes the undo.

Productivity Enhancement

Integrated software of almost any style enhances productivity.[5] Just as word processors improve productivity by minimizing retyping, integrated software has reduced the time it takes to perform tasks that require the use of several tools.

Several studies have been performed by outside groups that substantiate the assertion that Lisa enhances productivity. Seybold Publications Inc. did a comparison of Lisa, SuperCalc 3, Context MBA, and Lotus 1-2-3.[5] The task was to prepare an operating budget, which included spreadsheet calculations, making graphs, looking up information, and preparing a report. The timings did not include set-up time for the model, nor thinking time, consequently the timings do not represent the total time to complete the task, but only the time necessary to perform the four specific tests. The article reports that the total time it took was 27 minutes for a SuperCalc 3 user, 33 minutes for Context MBA, 47 minutes for Lotus 1-2-3, and 59 minutes for a Lisa user.

We couldn't understand how it took them so long to do the tasks using Lisa, so we looked into it. It turns out the users were experienced IBM PC users, and the users were inexperienced in using Lisa. We retimed the tests with an experienced Lisa user and found that it took 19 minutes. So the study should have shown it took 19 minutes for an experienced Lisa user, 27 minutes for a SuperCalc 3 user, 33 minutes for Context MBA, 47 minutes for Lotus 1-2-3, and 59 minutes for an inexperienced Lisa user. The disparity of time between the experienced Lisa user and the inexperienced Lisa user is that the inexperienced one was unaware of a mechanism for transferring data from LisaCalc to LisaGraph, and cut and pasted the data cell by cell. Another reason for the disparity was that the inexperienced user did not take advantage of background printing.

This study was chosen because it illustrates several things. The fact that an inexperienced Lisa user was in the ballpark for these tests shows that an inexperienced user can effectively get the job done (especially when the combination of set-up time and thinking time dominates). The other important point that this study illustrates is that a simple user interface leads people to some incorrect conclusions. People wrongly conclude that simple user interfaces are good only for simple things, and that features are not implemented. In the case of this study, the simple user interface led the users to believe that they had mastered the system, so they failed to look up functions in the manual.

SUMMARY

Lisa achieves integration in a variety of ways. It uses an intuitive model using familiar objects that permit direct interaction as opposed to indirect interaction with a command language. In contrast to some systems that always prompt for individual steps, Lisa's user interface supports users at every point on the learning curve. The uniform user interface also provides for transferrable learning; the user needs to learn how to edit, print, and file only once, and then can apply that knowledge throughout all applicaions. The combination of gradual and transferrable learning results in a system that is many times easier to use. This ease of use has made it possible for individuals to use tools to accomplish tasks that they could not before.

The editing model provides not only for editing information within a document, but also promotes the interchange of data among documents of similar or different types. This interchange is fostered by the ability to have several windows displayed concurrently. The ability to use documents concurrently and interchange data among them makes it possible to use several tools in a cooperative manner to perform a task that one tool alone could not accomplish. Since one of these windows can be connected to a remote computer, this permits exchange of data between mainframes and Lisa documents. In contrast to some systems, Lisa is an open system, permitting third parties to develop additional applications. Third-party-developed applications function in a manner that is similar to other Lisa applications with the uniform user interface, and are able to interchange data as well.

The combination of the uniform user interface coupled with multiple tools that operate concurrently, each of which can interchange data with others, provides an environment that increases office worker productivity. Workers are free to concentrate on their tasks, not on how to accomplish them.

REFERENCES

1. Goldberg, A., and D. Robson. *SMALLTALK-80 The Language and its Implementation.* Reading, Mass.: Addison-Wesley, 1983.
2. Zloof, M. "Query by Example." *AFIPS, Proceedings of the National Computer Conference* (Vol. 44), 1975, pp. 431–437.
3. Smith, D. C., C. Irby, R. Kimball, and E. Harslem. "The STAR User Interface." *AFIPS, Proceedings of the National Computer Conference* (Vol. 51), 1982, pp. 515–528.
4. Daniels, B. "Lisa's Alternative Operating System." *Computer Design,* 22 (1983), pp. 159–166.
5. Uttal, B. "The Best Software for Executives." *Fortune,* December 26, 1983, pp. 136–142.