Users and Personal Computers: Languages and Literacy, Costs and Benefits

Martin L. Ernst

Program on Information Resources Policy

Harvard University

Center for Information Policy Research

Cambridge, Massachusetts

A publication of the Program on Information Resources Policy.

Users and Personal Computers: Languages and Literacy, Costs and Benefits

Martin L. Ernst
January 1993, P-93-1

Martin L. Ernst is a research affiliate of the Program on Information Resources Policy, where he guides research into the influence of computer-related advances on skills both businesses and workers will require in the future. Prior to joining the Program, he was at Arthur D. Little, Inc., for 27 years, retiring as vice-president for Advanced Information Technologies. He is past president of the Operations Research Society of America.

Printing 5 4 3 2 1

January 1993

PROGRAM ON INFORMATION RESOURCES POLICY

Harvard University | **Center for Information Policy Research**

Affiliates

ABRH Consulting, Inc.
Advertising Mail Marketing Association
American Telephone & Telegraph Co.
Ameritech Corporation
Apple Computer, Inc.
Arthur D. Little, Inc.
Auerbach Publishers Inc.
Australian & Overseas Telecommunications Corp.
Bell Atlantic
Bell Canada
BellSouth Corporation
Centel Corporation
CMC Limited (India)
Commission of the European Communities
Communications Workers of America
Computer & Communications Industry Assoc.
Corning Inc.
Cox Enterprises, Inc.
Dialog Information Services, Inc.
DRI/McGraw Hill
European Parliament
France Telecom
Gartner Group, Inc.
GTE Corporation
Hitachi Research Institute (Japan)
IBM Corp.
IQ, Inc.
Information Gatekeepers, Inc.
Information Industry Association
International Data Corp.
International Monetary Fund
International Resource Development, Inc.
Invoco AB Gunnar Bergvall (Sweden)
I.T. Direction Ltd. (UK)
Japan Telecom Company
Kapor Family Foundation
Knowledge Industry Publications, Inc.
Korea Telecom
Lee Enterprises, Inc.
John and Mary R. Markle Foundation
Martin Marietta
McCaw Cellular Communications, Inc.
Mead Data Central
MITRE Corp.
National Telephone Cooperative Assoc.
The New York Times Co.
NEC Corp. (Japan)
Newspaper Association of America
Nippon Telegraph & Telephone Corp. (Japan)
Northeast Consulting Resources, Inc.
Northern Telecom
Nova Systems Inc.
NYNEX
Pacific Telesis Group
Philips Kommunikations (Netherlands)
Public Agenda Foundation
Puerto Rico Telephone Co.
Research Institute of Telecommunications and Economics (Japan)
RESEAU (Italy)
Revista Nacional de Telematica (Brazil)
Rhode Island Public Utilities Commission
Scaife Family Charitable Trusts
SEAT S.P.A. (Italy)
Siemens Corp.
Southam Inc. (Canada)
Southern New England Telecommunications Corp.
Southwestern Bell
Sprint Communications Company L.P.
State of California Public Utilities Commission
TEKNIBANK S.p.A. (Italy)
The College Board
Times Mirror Co.
Tribune Company
United States Government:
- Department of Commerce
  - National Telecommunications and Information Administration
- Department of Defense
  - National Defense University
- Department of Health and Human Services
  - National Library of Medicine
- Federal Communications Commission
- General Services Administration
- National Aeronautics and Space Administration
- National Security Agency
- U.S. General Accounting Office
- United States Postal Rate Commission

U S West

Acknowledgments

The author gratefully acknowledges the very kind help of the following people who reviewed and commented critically on the draft version of this report: John A. Arcate, David J. Berman, Keiichiro Kagiyama, Joost Kist, Peter B. Maggs, Vincent Mosco, Martyn F. Roetter, and Jerry Wasserman.

These reviewers and the Program's affiliates, however, are not responsible for or necessarily in agreement with the views expressed here, nor should they be blamed for any errors of fact or interpretation.

Executive Summary

The user beginning to work with a personal computer is starting a complicated operation that, examined closely, turns out to be largely a form of self-communication. Like all other forms of communication, this one requires an accepted language whose nature has an enormous influence on what can be communicated and how accurately and how effectively.

Computer languages, which come in many varieties, can be distinguished mainly by their purposes. Those of direct concern to users are the languages of operating systems (which provide means to control system resources and data flows) and of applications programs (which help users perform specific tasks).

A fundamental difficulty in working with computers is that almost all the languages important to users are different. Even among those intended for the same narrow purpose (for example, word processing), each producer of the application program will have a different set of commands (or, vocabulary) and a different way of combining them (or, grammar) to achieve a desired effect. Further, for major applications the "size" of each of these languages approaches that of a basic version of a spoken language. Before being able to benefit from having a PC, the user must learn one or more of these languages.

To counter this difficulty, computers can make use of two critical advantages: great speed and great flexibility in display of output. These capabilities are responsible for their success in fields—such as transaction and text processing—where computers have rapidly come to dominate operations. The same advantages also have been used to exploit other opportunities that have not received as much explicit attention, including:

- Hiding much of the inherent complexity of the system by instructions written in the layers of languages invisible to users
- Incorporating audio clues and simple manipulations of familiar image symbols into the operating system and application program languages, thus making these languages less arbitrary, more intuitive, and much easier to learn
- Encouraging user customization, experimentation, and simulation activities, all of which make the PC an effective vehicle for extending the user's existing skills and encouraging the learning of a wide range of new ones.

The penetration of personal computers beyond mainstream business applications will depend on user perceptions of the benefits received and the costs paid to obtain them. There are many classes of users, according to the amount and type of external support for PC operations easily available to them, and many forms in which user costs must be paid. When examined in detail, the most significant personal cost for users with a variety of purposes in mind for their PCs is learning a useful collection of application program languages to a level that permits their comfortable operation.

Such learning, itself an important information package, requires gaining knowledge of the substance (vocabulary and grammar) and acquiring skills in the processes (activities needed to apply the knowledge) associated with each particular computer programs. By exploiting the

skill-training capabilities of the computer more fully and in new ways, the user's learning process can made easier and more satisfying. However, the possibilities for new applications of interest, each with its own new language elements, are essentially unlimited, so, regardless of how easy the learning process becomes, the increase in types of applications imposes on users a need for continuous reevaluation of the costs and benefits of extending their range of use of their computers.

Over time, the skill-training and user experimentation capabilities make PCs a valuable participant in almost all processes of both formal and self-education as well as a significant tool for expressing literacy in the broadest sense. Given their growing importance in both education and in the expression of literacy, the amount of effort a user decides to commit to learning that extends use of PCs can be viewed not so much as an independent business decision but, rather, one to be integrated into the far broader set of personal decisions concerning the kind and level of general education each will seek.

Contents

Figures and Tables

Preface

In the turmoil and uncertainties that surround prospects for the computer industry in the early 1990s, a review of some of the fundamentals of computer operations can furnish guidance on how the industry may evolve. This report focuses on personally operated equipment and examines the following subjects:

- the basic nature of what users do when "computering"[1]—that is, working with a personal computer
- the critical role of "languages" in the process and the various types of languages
- the sources of unique advantages computers offer, relative to other means for information acquisition, processing, and distribution
- the costs in time, effort, and skills development users must pay to gain these advantages.

This examination leads to recognition that the user's interface with the equipment has even more significance than is often attributed to it, and suggests that the contributions computers can make to the general process of education are somewhat different from those usually claimed, with potentially greater breadth of application than anticipated.

---

[1]The word "computer" is too well established to be dropped in favor of an alternative, but an activity that *will not* provide the primary employment of personal types of computers in the future is "computing" (in any normal sense of the word)! Thus, the term "computering" is offered here deliberately.

Chapter One

How Computering Is Different

1.1 Computers and Languages

One way to view the actions of a user engaged in personal computer operations[2] is to consider them as being totally devoted to a stream of communications. The stream, however, is unusual in at least two respects. First, regardless of the intended audience for the results of the operations—which can be quite large—the immediate linkage is confined to two participants: the user and an inanimate, robotic entity. Second, in contrast to the common view of what "communication" is all about, much of the time the user is communicating primarily with him or herself.

Except when equipment fails, the communication will be flawless. Most of the time, the user will take the initiative by transmitting an instruction, or an item on which one or more instructions will operate. The computer will always understand what the user has sent (even though, if the user makes a mistake, the computer usually will not recognize it and will fail to comprehend what the user's *message was intended to mean*). The computer always will respond. The responses normally will indicate acceptance or compliance with the user input (even in cases where the user's input will fail to fulfill the user's intent) and often will present options among which the user should choose for proceeding further. Part of the response constitutes an indication of readiness to receive a new input. Far less frequently, the computer will signal that it cannot take any action in response to an input, for a given specific or general reason.

The self-communicating aspect of personal computing is easily overlooked, because it simply represents a new way of performing a familiar function. Every time a draft of any type

---

[2]For the purpose of this paper, personal computers and workstations can be treated as a single family of computers, and, for brevity, the abbreviation PCs will refer to this family. Workstations usually are more powerful, have more connectivity, are higher in price, and are marketed differently from PCs. For this study, the differences can be considered as amounting to workstation performance "lead times"—and at present rates of progress, PCs are never more than a few years behind workstations.

of document is prepared, the user is in a self-communicating mode, putting thoughts into a form that permits their inspection and improvement before they are considered ready for an intended purpose. Every time the analytical apparatus accessed through a modern personal computer is put to use, such as when preparing spreadsheets for a report, the user is similarly probing and testing to establish findings and implications that can lead to later action. In both cases, the operation could be executed (in theory, if not in practice) with pencil and paper—or on parchment with a quill pen, or, in a still more primitive period, by carefully trained memories that probably provided the means to perform at least some of the then current forms of the same functions. But the more self-contained nature of all of these earlier modes of working hid the communications processes that can be seen explicitly when a computer is employed.

The communications-based approach for examining the use of personal computers may seem forced, but its virtue is that it rapidly focuses attention on the basis of all communications—the languages employed. From the pheromones used by insects to the latest digital, multimedia, interactive compact disk systems, all types of communications require that, at some point, a signal representing or demonstrating an initiator's need or offering be created and somehow made available where it can be received and acted on. The sets of all such symbols and representations (the vocabularies), and the rules for combining them in an acceptable fashion (the grammars), constitute the languages employed by men and beasts—and by computers. However, there are ways in which this standard definition of language is too narrow for some purposes. As technology has increased the range of means by which suitable signals can be created and moved, the tools used for these purposes have become increasingly intertwined with the "languages" they were developed to support. During the latter half of the twentieth century, this process has had particularly important impacts.

Marshall McLuhan is widely known for his statement that the medium is the message. This implies, in the "bundling" terms developed and used by the Program on Information Resources Policy (PIRP),[3] that the format of a message becomes a major part of its substance. McLuhan's comment is far too strong—substance and format always are separable

---

[3]See, for example, Anthony G. Oettinger, *The Information Evolution: Building Blocks and Bursting Bundles* (Cambridge, Mass.: Harvard Univ., Program on Information Resources Policy, P-89-5, 1989).

at some level—but the format employed clearly has strong influences on not only the effectiveness of a message but also the character of the "language" used. A simple example is a person's vocabulary, which tends to be appreciably larger for reading than writing and larger for writing than speaking.

The term "language" is common in computer literature, but it usually is applied in an especially narrow sense. It most frequently refers to one of the different sets of symbols and combination rules (e.g., FORTRAN™, C™, Pascal™, etc.) used to create computer programs, including the large application packages that enable users to employ computers effectively even with only limited understanding of computer operations. In practice, however, as the more successful application programs[4] grow in functionality and coverage, some of them begin to assume many of the features of languages themselves. First, through the use of macro instructions, and continuing through independent production of "add-ons" and "overlays" (specialized programs that rely primarily on the application's more general capabilities to accomplish specific limited goals), the application can be used as a basis for creating what amount to new programs. At this stage, the application's instructions sometimes begin to be referred to as offering a "higher level language."

A more constructive approach is to regard the total interface between a user and a major application program as a special language that must be mastered if the package is to be employed effectively. For this purpose, the total interface must include both the instruction set for using the program and those aspects of the visual and audio presentation structure that influence operations through the interpretation guidance they provide and the action options they make available.

The language-based approach offers two advantages. First, it provides a good perspective on the learning tasks users face. The size of the "vocabulary" of a major application, such as WordPerfect 5.1™ or Lotus 1-2-3™ (Rel. 3.1), and the complexity of its "grammar," are not

---

[4]The terms "application" and "program" are often used interchangeably. In this report, however, "applications" will refer to specific types of tasks (preparation of text, spreadsheet analyses, income tax calculations, etc.), independent of whether computers are involved, while "programs" will refer to particular packages (such as WordPerfect 5.0™, LOTUS 1-2-3™, etc.) designed to enable computers to help a user perform those tasks.

too different in magnitude from those of the most basic versions of spoken languages.[5] Minimum use of the application requires what would correspond to "tourist" lingual ability: ability to, say, order a meal, find a rest room, and instruct a taxi to go to a well-known destination.

The second advantage comes from inclusion of the visual (and sometimes audio) elements incorporated into the computer-application combination. The visual-audio features call attention to a unique aspect of computers: by combining their speed and the flexibility of their output display, computers can be programmed not only to provide user transparency through layers of inherent complexity, but also to present and offer means to control the computer's operator by manipulating simple symbols of easily recognized real objects—such as images or icons of files and papers on a desk, or the layout of a page of text or diagrams. This capability, usually made available through what is termed a Graphical User Interface (GUI), can be regarded, itself, as a special type of language, although one whose main value arises only when employed with another (application program) "language." GUIs can give an air of familiarity to many kinds of command operations, making them extremely easy and natural to learn; these features make them of enormous value to beginners and less skilled users.[6]

The analogy between spoken textual languages and the "languages" of computers was described in a previous Program report[7]. Particular attention in that document was devoted to identifying the reasons for the dramatic growth that has arisen, over time, in the capabilities, the sizes, and the *inherent* complexity of typical major application programs. Part of this

---

[5]According to *The American Heritage Dictionary* (1969), "Basic English," which received a lot of attention in the 1950s, provides a copyrighted, simplified version of the English language, using a vocabulary of about 850 words. This is not much larger than the list of all commands and options characteristic of a major application, especially if Add-Ons and common Macros are included.

[6]The move to supplement GUIs with capabilities for entering and later retrieving (but, in the early 1990s, not yet interpreting or processing) handwritten and spoken entries is already underway. When effective, the results may ease some user learning burdens (to the extent that the potential gain is not diverted into making the applications larger and more complicated!) at the cost of more design complexity.

[7]Martin L. Ernst, *The Personal Computer: Growth Patterns, Limits, and New Frontiers* (Cambridge, Mass.: Harvard Univ., Program on Information Resources Policy, P-91-6, 1991).

growth process produces a parallel increase in the application's vocabulary size and in the inherent complexity of its grammar.

The reasons for growth in application sizes mainly involved efforts to meet different perceived needs and interests of the market. They included: expanding the functional coverage and abilities of the application, maintaining compatibility among successive program versions, providing interfaces to both a growing variety of peripherals and competing or complementary software packages, simplifying the user's work, and providing many other potential aids to effective application use. Although the increasing complexity of the growing program package sets limits on what can be done at any given time, these limits will be temporary. The market-based incentives for growth are strong, and continuing improvements in the performance of electronic components can provide the computer speed and memory needed for the expansion process. The complexity problem can be alleviated, *from the user's point of view*, by devoting a portion of the increases in electronic capabilities to "hiding" some of the inherent complexity (rather than using all of the new capabilities to increase functionality). Although the hiding operation throws a burden of complexity management on hardware and software designers, it helps meet a critical user requirement for achieving a new increment of growth in the marketplace.

Another conclusion of the work was that, within carefully established contexts, computers gradually are becoming capable of processing words in ways that will make the computers appear to outsiders as though they understand the symbolic content of the text being processed. One way these abilities could be exploited would be to improve user management of information flows by subjecting incoming information streams to a variety of extensions of existing computer search and match operations. However, to establish both the effective context and the controls needed to benefit from these operations, the user will have to have great skills with words and in understanding the capabilities and limitations of language—what amounts to demonstrating a high level of literacy in a very traditional sense.

Literacy and technology are intimately related, since the former is concerned with both a sound understanding of the possibilities and limits of a language and skills in using the tools available for expressing (e.g., creating) and communicating in that language. Because technology is the primary source of new tools that can change and expand the ways a society uses to express and/or communicate in a given language—or even create new languages—it plays a major role with regard to both how languages are used and the nature of literacy at a given time in history. The suggestion made above, to the effect that to be an effective user of personal computers may in the future require a high level of a classical type of literacy, points to at least a partial convergence between literacy in the use of the very newest set of tools available and literacy in the use of some very old ones.

1.2 How New Technologies Spread

To appreciate better just what is happening in the world of computers, it is useful to examine some of the patterns that arise whenever applications of a major new technology begin to appear in the marketplace. Usually one of the first impacts is an increase in the complexity of the total systems in which the technology is being applied, because the new technical devices will be relatively poorly understood and untested. As a result, both the products created with the new capabilities and the mechanisms for their use are apt to be quite clumsy. Furthermore, the new products are liable to be unfamiliar to users; this, by itself, can create a perception of their being complex, whether justified or not. Finally, the early applications of a new technology usually comprise near replicas of only part of what was done by a prior device and the system within which it was used; this situation often leads to the co-existence of both old and new systems. All these factors will amplify the costs and the difficulties of early users and increase the perceived, and often the actual, complexity of operating with the new applications.

Sometimes the added complexity is widely visible throughout society, and sometimes not. The most obvious instances of highly visible extra complexity arise when, to exploit the technology, a lot of users must make the effort to learn how to use a variety of new tools. With the spread of printing, for example, many users had to learn to read and to spell, and (to a lesser extent) to write with a quill pen. Later, the care and use of pencils and pens required another set of new skills,[8] as did the typewriter and the pocket calculator. These are

---

[8]The perceived complexity of these (now apparently simple) skills when they were first needed should not be underestimated. In *The Pencil* (1990), Henry Petroski notes: "Manufacturers of the first mechanical pencils certainly seemed to take little for granted and published 'directions for use' right in their advertisements: 'Hold the two milled edges between the finger and thumb of the left hand. Turn the case with the other hand to the right,

easily noticed new needs. Other requirements, like the myriad skills associated with all the tools in printing operations, as printing technology evolved over time, incorporated much more complexity; however, the need to cope with this complexity was confined to a specialized sector of the population and was thereby hidden from the direct view of those who created or used print products.

As time passes and use of a new technology spreads, the inherent complexity of the applications derived from it will tend to grow, but an increasing fraction of the complexity will be made invisible to the end-users of the technology. The automobile furnishes a good example; few present drivers have, and even fewer need, the degree of understanding of what goes on under the hood of a car that was almost a prerequisite for driving, say, 70 years ago. A whole set of controls—starter crank, choke, spark timing controls, manually controlled gears, etc.—either have disappeared or can be avoided at will. But the inherent (and greatly increased!) complexity of the car hasn't disappeared; the need to deal with it has simply been transferred from the user to the layers of designers involved in the production of the car and its many components.

As described in **section 1.1**, the same process has been underway with personal computers. As the needs and desires of users—and, even more important, of *potential users*—became better understood, market forces drove both hardware and software producers to find ways to meet these needs. And, because easier operation of computers and associated software was a major desire of most users, the producers were forced to accept the incorporation of ever greater design complexity in their products in order to make their offerings appear less complicated in the marketplace.

For this complexity-hiding process to continue, the pace of advance in the underlying technologies must remain strong; these advances provide the resources both to create new functionality and new uses—so that the potential market can keep expanding—and simultaneously undertake needed simplification actions. Stability in the underlying technologies will slow the rate of change in the products being built, making it easier for

---

and the lead will be propelled....'" The instructions continue for three to four more lines at the same level of detail.

users to learn how to use them; it will, however, also gradually stifle the ability to keep performing the magical hiding act. Finally, complexity-hiding has its own risks; unless all possible desired uses of an application can be foreseen during the design process, the means to perform some valuable operations may be hidden inadvertently. An important feature of good complexity-hiding is that means are provided that enable the more skilled users to by-pass or dig below the camouflage and modify portions of the system (such as its software) to meet personal needs better.[9] This process also provides a source of ideas for the additions and improvement in later versions of the system.

The requirements for new skills and the management of complexity are only part of the story; adequate benefits must be received or a prospective new technology will languish unused. One form in which benefits are received is as decreases in the need to learn those older skills that are made obsolescent. Thus, in the information fields, writing and reading skills greatly lowered the need for memory skills; word processing has almost eliminated need for the ability to type without error; and the hand calculator has decreased requirements to learn how to use logarithm tables or manually calculate the square roots of numbers. Because these skills may already have been learned, the benefits they provide often will concern "sunk costs" as far as a major part of the initial user population is concerned; while important, these benefits normally have small impacts compared with benefits on the positive side. In the information field, the positive benefits can come in many dimensions—quantity and quality of

---

[9]Whenever complexity has to be dealt with, providers must choose a position somewhere between two extreme strategies: to hide only some of the complexity, leaving the rest for the user to handle, or to seek to hide essentially all of it. The latter policy, most commonly found among popular consumer goods, has the obvious advantage of appealing to markets that require simplicity, but it often eliminates opportunities for some users to make important choices and customization decisions. This policy may be difficult to maintain during periods of rapid technical advances, especially if the product is not cheap enough to throw away after limited use or not easy and economical to trade up.

This paper will usually take an intermediate position, but always one where the more skilled users *can exercise* considerable choice and *can make* fairly significant changes (or supplements) to standard system and application settings and other features. This position appears to represent the one adopted by most major PC providers in the early 1990s. Larger computers tend to place more burdens on users, while the maximum simplicity approach is limited to specialized applications; some examples are game playing computers and electronic typewriters. The tension in the situation is well illustrated by Nintendo™, which started as a game-playing machine and later grew in capabilities and began to encompass other niche functions.

outputs, variety of formats that can be used, scope of substance coverage, speed of creation and delivery, security and survival capabilities of the storage media, the range of social participation made possible, and many others.

For major technologies, the full range of new positive advantages tends to be identified and exploited slowly, with the pace quickening only after the number of skilled users reaches a critical mass. The extent and speed of adoption also depend on many factors other than growth in the functionality of applications and the quality of their complexity management. A broad range of requirements for new capabilities and features usually will be identified through market experience. These requirements will go far beyond pure technical matters to include needs in the social and cultural, economic, business practices, and legal and regulatory spheres. As these factors are considered, and means to service them evolve, there almost always will be a growth in the total complexity of the resulting "system" (e.g., the design, manufacturing, marketing, physical distribution, servicing complex). If outputs of the system are then to be marketed successfully, an increasing fraction of this nontechnical complexity also must be hidden from the large majority of users. As noted, this process may continue for a long time—it can go on until the underlying technology and its primary derivatives have spread and stabilized. The implications of this diffusion process are that, if the applications of interest require wide-spread participation by the population as a whole (rather than operation only by specialized groups), the process may require a generation or more and may not be achieved until the majority of new users have learned the needed skills as children.

1.3 What Computers Offer—and What They Cost

Where do personal computers fit in the scheme of things discussed so far? To start, their communications (at least in western nations) are based on a set of about a hundred characters—the number of keys on a typical full keyboard. As with normal languages, these characters can be combined to create any desired number of "words." The most meaningful words for using computers, however, are not natural-language ones (although the latter often form the large majority of user keystrokes, because they constitute most of the data or information input), but the special "words" that provide commands and/or inquiries to the computer. Although theoretically unlimited in number, in practice there will almost always be no more than the mid-hundreds of this type of formally specified words. One factor setting this limit is the need to achieve the simplicity and "user friendly" interface the market demands. The simplicity requirement implies one must be able to access most of the command and control "words" from memory, or by a (not overly cumbersome) system of icons and/or menus.

But this requirement is not the complete story. As mentioned earlier, most important applications programs have had Add-Ons, Overlays, and other forms of extensions built to work with them; these inevitably have additional commands, some of them important to learn. Also, the more skilled users can build (and others can buy) sets of Macros—simple new "words" created by combining what can be fairly long strings of existing "words." These amount to new, personally designed, relatively complicated commands that can be issued with a single, brief instruction. Other forms of "personalization" often are possible, ranging from those dependent primarily on natural-language skills (such as building lists of personal associative "synonyms" to be used in a search program) to others that require a high level of computer and software skills.

The above discussion illustrates the close intertwining of the great strengths and the important weaknesses of computers. At their best, computers place enormous capabilities in the hands of users. Almost any major application program enables users to access and employ types and levels of expertise that are, in total, far beyond their normal, unaided reach. A good word processor makes it possible for a writer (or a secretary) to correct spelling and grammar painlessly, practice the arts of editing, page composition, printing, and mini-publishing, and perform a variety of distribution operations.

Of equal importance is the process by which this expertise can be acquired. This is a task in which the computer's flexibility of operation and the plasticity of its output can be put to good purposes. Once competence in a central program (e.g., word processing) has been achieved, many incremental capabilities (such as editing and publishing skills) can be gained by "playing with the possibilities"—combining lots of trial-and-error practice with limited amounts of reading and more formal training. Making a major contribution to this activity are the transparency and simulation capabilities of computers described earlier. Some combination of training, reading, receiving continuing guidance from the learning system (in this case, the application program itself), and practice is needed for almost all types of learning, but the process tends to be far more difficult, expensive, and time-consuming in the more rigid traditional learning environment than when it involves a suitable computer application.[10] The end result is that, at its best, there can be real pleasure in learning to use new computer capabilities. Korzybski once noted that, compared with plants, animals' mobility made them "space-binding" and that, compared with animals, humans' memory and communications abilities made them "time-binding."[11] In these terms, and relative to pre-computer days, the speed, packaging, and learning approach capabilities offered by computers and their programs enable people to become far better at "skill-binding."

Of course, a price must be paid: one must learn and understand all those new "words"! Learning several hundred new words by heart is no problem at all to a young child in the process of building a vocabulary, and it's not so difficult a challenge to most adults. As just noted, if the process is well designed, it often can be fun. However, we all live under time pressures; so, if the effort required is deemed large, most people won't want to do the job repeatedly. The learning task will make sense only if there is an acceptable combination of the value (*i*) of the output made possible, and (*ii*) of the level of utility and stability in employment of the "words," once they have been learned. This, of course, is the rub; at the present stage of development, each application for each different family of "compatible" computers has its own "language." Often these languages use the same "words" (sometimes with the same and sometimes with a different "grammar") for entirely different purposes. Further, although the process may be a slow one, the words used for a given application are likely to change over time, as new features are added which force changes in menu structures and reallocation of the meanings of keystroke combinations.

Much of the above is inevitable. Different computer applications require different sets of user actions to accomplish their purposes. They involve different skills and background knowledge, employ different vocabularies, and have their own special jargons; and all of these change over time. The situation is not unique; comparable problems arise (although

---

[10]Possible large advantages of the traditional learning experience are not covered here. The above comments apply fairly narrowly and should not be construed as a general attack on traditional approaches or unconditional support of computer-based methods.

[11]Alfred Korzybski, *Science and Sanity: An Introduction to Non-Aristotelian Systems and General Semantics* (Lancaster, Pa: Int'l Non-Aristotelian Library Pub. Co., 1933).

usually at a slower pace) when ordinary languages are employed in different disciplines. In some ways computers can make the general learning process easier.

For example, the use of GUIs not only makes initial learning easier but also has tended to force a certain amount of de facto cross-application standardization. Largely because all its application programs must operate with a standard, proprietary GUI, programs written for Macintosh™ computers share more command actions than those written for the IBM™ and compatibles family.[12]

However, there still are limits to what can be done with the methods available at the start of the 1990s. Even for the Macintosh, there is essentially no means whereby one can be assured of an ability to transfer a significant fraction of the command skills learned for one major program immediately to another. The final blow is that the situation will likely become worse in the future. The variety of new user interfaces being developed, and the steady (albeit slow) incorporation and integration of multiple different communications elements within a single output (e.g., some or all of numbers, tables, text, graphics, images, sound, and video), practically assure at least some continuing instability in command languages—and thus more material for the "fully skilled" user to learn, even if learning how to handle each component is simplified by complexity-hiding.

One obvious result of these language characteristics is the creation of a major barrier to the spread and use of personal computers. Where a single application is sufficiently important to justify the equipment acquisition and user training costs involved, PCs are widely employed and often constitute the dominant means for performing that type of work. Where justification requires active employment in multiple types of uses, personal computers have been relatively unsuccessful—even when good quality programs for the individual applications of interest have been available. Thus, the current scene shows the PC occupying many rather narrow

---

[12]There is a downside to this observation. The legal system for protection of intellectual property can easily end up *increasing* users' difficulties if it too broadly supports potentially transferable interface features as subject to copyright. For a discussion of the many issues involved, see Anne W. Branscomb, *Nurturing Creativity in a Competitive Global Economy: Intellectual Property and New Technologies* (Cambridge, Mass.: Harvard Univ., Program on Information Resources Policy, P-88-4, 1988).

niches of applications and having fairly limited impacts on the broader uses society makes of information.

The problem of multiple, unstable languages[13] that users must learn, if they are to employ their computers for a variety of tasks is only one of many factors slowing the spread of PCs. In a previous report,[14] we identified and examined a large number of such barriers. These included several purely technical weaknesses, a number of user concerns (other than those associated with learning efforts), problems arising from industry attitudes and practices, some important new infrastructure requirements, and various social, legal, and regulatory questions needing resolution or clarification. The user "languages" problem—or, more broadly, the closely related *total user "investment"* problem, focused on in the next section—provides only part of the full picture, albeit a critical part.

The prize is important. Computering *is* different. The packaging and display elements that enable it to be "skill binding" also enable it to be multimedia and multitool-using, and, at its best, a device on which it can be fun to learn new skills, a means for many types of recreation, a support for beginners and experts in businesses, professions, sports, music, art, and crafts, and on and on, into areas not yet thought of! But these are still mostly only partly realized capabilities and distant goals, and they will remain so as long as the price users face is perceived as so high. An understanding of these perceptions, and the reasons for them, can be valuable for actions in many different arenas.

---

[13]Neither multiplicity nor instability is unique to the languages of computers. Natural languages also are continually changing, even when efforts are made to avoid change. Each language has its own variants and jargons, used to distinguish social classes and locations of origin, to fit particular environments and activities, and to support and/or "protect" special skills and professions. What is different in the computer environment are the extent of multiplicity, speed of change, and the potential high cost of failing to keep up to date once a degree of dependence on the use of PCs has been established. The pace may decrease with industry maturation, but not rapidly.

[14]Martin L. Ernst, *Electronic-Print Competition: Determinants of the Potential for Major Change* (Cambridge, Mass: Harvard Univ., Program on Information Resources Policy, P-89-4, 1989).

Chapter Two

What You Pay to Be a User

2.1 The Setting

2.1.1 Users, Cycles, and Decisions

A fundamental feature of the PC/workstation environment is that all users are not equal! They can differ in many ways, but a particularly important difference is the level of external support users receive for conducting computer operations. At one extreme are members of a "pampered" set who, because of their positions or the nature of their jobs, have their every computer need attended to. Hardware, software, and maintenance are provided at no cost to them; formal training is available (or even forced on them) when needed; and experts are on hand to deal with questions that arise. At the other extreme are individuals who receive no support whatever. Hardware, software, maintenance, and training must be purchased with after-tax dollars; access to expert help or assistance is nonexistent; and sometimes the result is a computer retired to collecting dust in a closet. In between are many intermediate cases. The situation is summarized in **Table 1**, where a sample of user segments is characterized in terms of the financial, special training, and expert advice support typically available to them.

The support systems covered by **Table 1** are not employed on a "one-time" basis—that is, only when PCs are first introduced. Largely because of continuing rapid advances in the underlying technologies, support must be a continuing activity. An appropriate way to examine this type of process is in terms of a procurement cycle, illustrated by the three stages shown in **Table 2**:

- **Initial Acquisition**, where a full set of hardware (i.e., computer and main peripherals) and software (operating systems and primary applications) must be procured.

- **Improvements**, which can vary in importance (depending partly on the anticipated duration of the full cycle), and include selective aspects of all the original (and, sometimes, certain new) types of hardware and software.

Table 1

Types of Users and Support They Receive

USER CATEGORY	FORMS AND LEVELS OF SUPPORT			MAIN TYPE OF USERS IN CATEGORY
	Financial Assistance	**Formal Training**	**Access to Expertise**	
Lonely	None	Minimal	Minimal	Home Users who have no other serious contact with computers
Befriended	None to little	Minimal to little	Considerable	Home Users who have contacts with skilled computer users at work
Suppported	Major to complete	Considerable	Considerable	Users in working environments where they operate with computers
Expert				
Professional	Major to complete	Major	Extensive	Computer specialist
"Afficionado"	Varies	Varies	Extensive	

- **Major Re-equipment**, which normally will amount to replacement of most or all of the existing equipment and a major upgrade of software. In some cases, requirements to operate new software will force the acquisition of new hardware; in others, software purchases will be the follower, encouraged by the increased capabilities that come with the new hardware. In either case, the re-equipped configuration plays the role of "Initial Acquisition" for the next cycle.

The entries for the three stages in **Table 2** are taken from the main offerings of vendors during the 1980s and very early 1990s. The nature of the offerings in the next decade will almost certainly be different—perhaps radically so—but the general cyclical process is unlikely to change so long as technological advances keep offering the computer industry both opportunities and incentives to make rapid changes in equipment and software.

Table 2

Personal Procurement Cycle: 1980s and Early 1990s

ITEM	ACQUISITION	IMPROVEMENT	RE–EQUIPMENT
Hardware	Computer	Add: Internal RAM Internal hard disk Coprocessor Graphics card	New computer with: Greater power New features
	Peripherals	Add: Disk memory capacity Modem Color monitor CD ROM drive Fax driver	New peripherals with: Greater power Better quality Other features
Software	Operating and support systems	Upgrade to new version of operating system Add new utilities programs	Major upgrade of operating system, or change to new type
	Applications programs*	Upgrade current programs to newer versions Acquire add-ons to current programs Acquire new types of programs	Major upgrades or changes of programs Acquire new programs that older hardware cannot

*Applications are tasks the user wants to perform. Programs enable the computer to assist the user.

Although all active users will undergo some form of the procurement cycle, there will be differences among the various segments in many of the details of their cycles. First, there will be a spread in the durations of the cycles typical of the various user categories. For example, in most situations, those in the Expert category will have to be well equipped if they are to perform their functions or follow their interests effectively. Professionals usually will be given (and Aficionados somehow will acquire!) the latest and most powerful hardware and software available within relatively generous budget limits. With the pace of technical change characteristic of the 1980s, staying up to date required re-equipping on the basis of short time cycles, of the order of two to three years' duration. The short cycle times, in turn, tended to make the improvement portions of the cycles relatively unimportant.

During the 1980s, the Supported group normally had longer cycles—say, three to four years—although some segments kept their original equipment well beyond this period. The longest cycles have been those of the Befriended and Lonely groups, where seven to ten years offers a reasonable range (but with some drop-outs who never improved or re-equipped). Here, the long cycle time made hardware improvements a bigger factor. Some of these had to be made if users wanted to take advantage of specific new software products that offered benefits too valuable to ignore but that also required upgrading their existing hardware.

A second difference concerns the acquisition decision process. Early in the history of PCs, there was a brief period of near anarchy in corporate PC hardware and software purchasing procedures. Since then, we have returned to the normal business practice, where those who do the paying make (or approve) most of the decisions that authorize how much is to be spent and what is to be acquired. In the work environment, organizations almost always provide any needed computer resources; accordingly, among their employees either no users, or only a few (mostly professionals with special needs), are likely to have more than a limited choice of either the hardware or the software they will be operating. Effectively, most users in the Supported segment will have to take what their organizations offer. If any of these users decides to acquire a home set-up, there often will be strong advantages to having a high degree of office-home equipment compatibility. In this way, the office selection sometimes can force the choice of home equipment and software—even though at home, the users may be operating in a largely self-supporting way. With this exception, the extent of choice tends to go up as support goes down. Unfortunately, budgets and training opportunities almost certainly will not go up as fast as support declines, so the advantages that can be taken of the greater choice may be quite limited!

Finally, depending on many factors (positions, functions, interests and experience levels of individual users, types of business activities for which the computers will be used, and many other situational features), there will be large differences in exactly what is acquired by or for users in each category. For some occupations, equipment must incorporate excellent communications capabilities and permit intensive real-time operations. For others, computers with very fast numerical processing speeds and complex analytical software may be emphasized. A very large group of users will be engaged in fairly prosaic, highly structured processing activities. Managers of typical installations handling this type of work might describe their basic needs as comprising: generally fast equipment with large capacity; software of narrow scope, requiring neither frequent nor rapid changes; low system acquisition and operating costs; and high levels of output accuracy and system reliability. Still another group, principally a portion of the users of home and hobby PCs, will weigh costs very heavily and will seek great simplicity and good documentation in their software, to balance their lack of both personal skills and easy access to more experienced users.

The patterns of all these decision procedures, and the factors that influence them, have had enormous impacts on computer industry markets. Most of these impacts have been recognized explicitly for a long time and are exploited routinely by marketers. The patterns also lead to major impacts on the nature of the tasks faced by all types of individual users of computers, and many of these, too, have been widely recognized. However, because most individual users have had no direct role in the acquisition decision processes of the *larger market segments*, their needs have usually been considered only in rather broad, nonexplicit ways.

2.1.2 Areas of Investment

To establish a more analytical perspective, we can start by simply listing the investments of time and effort that *all users* must make (or, for some items, have made for them) to operate their PC/workstation systems effectively—with comfort and confidence. This is done in **Table 3**, using the procurement cycle pattern described above. Although most of the items in the table are quite general, where specifics are used they apply to the environment of the late 1980s and very early 1990s.

The essence of **Table 3** is quite simple. First, and independent of the applications that may be used, the user must learn certain capabilities of the new computer and software. The user (or the supporting staff) must then plan and implement a variety of organizational structures and features to ease the process of maintaining effective control of data and operations in the future. Thereafter, specific applications programs must be learned to a level

Table 3

Required User Investments of Time and Effort: 1980s and Early 1990s

<table>
<tr><th>FOR INITIAL ACQUISITION</th><th>FOR IMPROVEMENTS (As Relevant)</th><th>FOR MAJOR RE-EQUIPMENT</th></tr>
<tr><td>1. Learn hardware connections, switch settings, set-up selections, general operating procedures, physical control tools, trouble indicators, etc.</td><td>Learn changes and additions</td><td>As for Initial Acquisition but requiring less effort</td></tr>
<tr><td>2. Learn operating system, start-up and configuration controls, user interface management, and support programs (utilities, diagnostics, security measures)</td><td>Learn changes and additions</td><td>Generally as for Improvements but requiring higher levels of effort for learning, installation, and personalization activites</td></tr>
<tr><td>3. Design initial organization for operations — directory and file structures, personal access menus, start-up procedures, document control, back-up procedures, etc.</td><td>Change organization design with experience, as applications are added or changed; implement changes</td><td rowspan="6">Effort can approach that required for Initial Acquisition: benefits from increased experience since Acquisition are partly balanced by:
➤ time pressures: user is more dependent on computer than before and is at a disadvantage (sometimes serious) until transition to the new system is complete
➤ increasingly broader applications and more complete programs may require users to learn new skills
➤ threatened loss of access to critical data unless special measures are taken for and during file transfer
➤ loss of, or need to redo, personalization features and macros if these are not directly transferable</td></tr>
<tr><td>4. Learn characteristics and commands, installation requirements, and key parameter settings of applications</td><td>Learn characteristics of new applications and changes and additions to older applications</td></tr>
<tr><td>5. Implement user's plan (Item 3) and install applications</td><td>Install new applications; "uninstall" or implement changes to older ones</td></tr>
<tr><td>6. Learn additional details of application programs as needed</td><td>Learn details of new applications</td></tr>
<tr><td>7. With experience, personalize applications and interfaces using:
➤ batch programs, macros
➤ special parameter settings to optimize system
➤ menus and other user-controllable operating aids</td><td>Continue personalization of both old and new applications:
➤ build and/or transfer macros and batch programs
➤ reset parameters
➤ build and/or transfer menus and other aids</td></tr>
<tr><td>8. Input or transfer necessary or useful fixed and "starting" data or information</td><td>Input new data and information; arrange to backup and transfer older materials, taking special measures to retain access to critical items not transferable to newer equipment or programs</td></tr>
<tr><td>9. Begin to investige and evaluate hardware and software for possible future aquisition</td><td>Continue to investigate both new hardware and software possiblities and the relationships between the two types of acquisitions</td><td>Continue to investigate new acquisitions emphasizing those that re-equiping makes feasible</td></tr>
</table>

that meets either personal or organization criteria. At some point, the users (or the experts serving their organizations) must review and evaluate possible additions or changes in hardware, software, and the operating environment she/he (or they) have created. This process must be repeated whenever a change is made, with the larger changes like re-equipping calling for many more user efforts than smaller changes.

At a more detailed level, the main capabilities and features to be mastered can be summarized as[15]:

- Basic computer hardware operations—equipment connection and set-up; switch settings as needed; general operating procedures and trouble indicators; and physical control manipulation (with cursors or a mouse).
- Basic computer software operations—operating system; start-up and configuration control procedures; user interface management; security arrangements (as appropriate); and basic hardware utility and diagnostic programs.
- System organization—directory, file, and indexing organization; personal access menus; documentation control procedures; back-up plans and procedures; and sharing/emergency access arrangements.
- Applications—program characteristics; commands, menus, etc. employed and the results they produce (to the desired or required level of detail); installation and set-up; special start-up and back-up arrangements (as needed or desired); and file transfers from older equipment or programs (if applicable), including creation of suitable back-ups for the transfer process.
- Personalization of basic set-up, and of each application—batch programs and macros; special parameter settings for system optimization; and customization of menus and other user-controlled features.
- Growth path—possible hardware and software modifications, improvements and acquisitions.

When viewing either the above or **Table 3**, considerations of exactly who does what are important. In the Supported category, many of the activities will be undertaken *for* the users, rather than *by* them. In some cases, the users will perform a limited amount of customization

---

[15]This list is essentially the same as in **Table 3**, but, to give a better sense of timing and sequence, here material is broken into smaller steps and ordered somewhat differently.

with regard to how they perform their work; in other situations, this will not be practical (or may not be permitted). In all cases, users will need to learn, or be trained to conduct, some general computer operations on their work system and to employ effectively specific application programs. Normally, users will not need to pay for training. However, many job openings are available only to those who already have basic skills in using a computer—and sometimes only to those with skills in the specific program version of the specific application used by the employing organization! In other words, some important training may be available only to persons already on board.[16]

An obvious reason for this policy is that most organizations consider training expensive, especially for staff who draw full pay during the training period and who, for a time thereafter, will work at less than full efficiency as their new skills are developed. These feelings have been widely expressed, by both users and producers of computer resources: "The real cost of software is not in the package but in the price of training"[17]; and "It's becoming apparent that the real cost is not the hardware or even the software. The real cost is teaching the user."[18]

At the opposite end of the spectrum, the Lonely user has an almost entirely different point of view. Paying for all goods and services with after-tax dollars means that costs will loom large, while time will be valued on a different scale. Depending on the user's attitude towards the computer, and on the total time demands that must be faced, the value of time can range from nearly zero to a level where it competes with money costs in importance. In almost all situations, these users will have higher (sometimes much higher) time requirements than

---

[16]For detailed examples of how requirements for prior training change as new technologies penetrate the commercial markets, see Benjamin M. Compaine, *New Literacy Indicators* (Cambridge, Mass.: Harvard Univ., Program on Information Resources Policy, 1986).

[17]Wayne Maples, Information Center Consultant at the Federal Reserve Bank, Dallas, quoted in D. Churbuck and B. Freedman, "Suits Against *1-2-3* Imitators May Have Wide User Impact," *PC Week* (20 Jan. 1987), **125**, 130. Cited by Richard A. Magnan in *Software User Interfaces Compatibility and Copyright After Lotus Corporation v. Paperback Software International* (Cambridge, Mass.: Program on Information Resources Policy, Harvard University, 1992).

[18]Interview with John Sculley, President and Chair, Apple Computer Corp., "On Fitting into the IBM World of Computing," *Personal Computing* (April 1986), **145**, 147; cited by Magnan, *supra*, note 17.

Supported users for conducting a computer-related task, because they will have to handle all activities entirely themselves, without substantial outside help. Advice and assistance are, by definition, almost totally absent; training, like equipment and software, must be bought with after-tax income. Under these conditions, even simple tasks can grow to apparently monumental proportions.

The Lonely user category is an extreme one. Members start out somewhat in the position of a reader, living without access to a library, who must purchase all reading materials—but who finds that everything the local bookstore carries is written in a foreign language! Because of its definition, this category almost certainly will be one with only a small population. A less extreme category, and a larger one, is the Befriended group. This group gets little financial support but does have access—through work, personal acquaintances, a local computer club, or some equivalent—to advice and assistance and, perhaps, even to some low-cost training opportunities. Nevertheless, the conditions that cause the Lonely segment to be small are important, since their situation shows, in exaggerated form, a set of user difficulties that constitute one of the bottlenecks to a successful future for PCs.

2.2 The Structure of User Investments

The items listed in **Table 3** do not easily lend themselves to a quantitative analysis. While samples of dollar and time costs could be collected concerning many of them, the interpretation of these data would be clouded by two major ingredients of the situation—variability and volatility. The factors that can produce variability in the data are legion. They start with the category of the user on whom data is being collected and go on to include: user level of experience within the category; type of PC/workstation being used; prior user experience with this (and other) equipment; the particular programs being used (in some cases, down to the level of specific program versions); the number of different programs typically operated by the user during the period for which data were collected, and their use frequency patterns; the numbers and kinds of functions within the various applications that were utilized (as this characteristic determined the required depth of the user's understanding of the program); the stability over time of some of the above; and many others. For meaningful quantitative analysis, the contribution of many of these factors to variations in the user investments of effort would need to be isolated; and the difficulties of doing so are obvious.

Volatility considerations are almost as far-ranging. Computers and software have been changing at a tremendous rate. This has made it possible to ease or eliminate some types of user problems, through improving software design and documentation, increasing equipment and software reliability, and identifying and solving specific user difficulties. At the same time, other problems were created by the greater ambitions and inherent complexity of the more recent products. Complicating all of these factors, the nature and attitudes of the population of users have changed fairly rapidly as an increasing fraction of them gained considerable experience with computers early during their education. As a result, there has been little overall stability—and almost any quantitative data on a specific type of user investment of effort can become obsolete or irrelevant almost as fast as they are collected.

2.2.1 The Critical Investment Area

While the above considerations limit the types of appropriate analyses, they do not block our ability to gain a better understanding of the structure of the general user learning problem. A starting point is recognition that the critical investments of time and effort users face are involved in learning the various applications and, to a lesser extent, the operating and utility management programs. This is illustrated in **Table 4**, where the items from **Table 3** are listed in detail and then characterized in terms of several attributes of use. As noted on the table, the contents apply to the turn of the 1990s. While many entries here are essentially universal, and time-independent for the foreseeable future, their relative difficulty[19] and importance will vary with technical and other changes.

A number of items in **Table 4** are used very infrequently, almost all of the uses arising when getting started or when changing equipment or software. Although this set of items, termed *Enabling* activities, may be bewildering or frightening when first encountered, usually they take little time to apply and are inherently simple. Also helpful, by the early 1990s, most industry providers had learned how to simplify the more critical operations that must be

[19]As described in note 9, providers have the option of increasing the simplicity of the activities listed in **Table 4**, by hiding the complexity (when practical), limiting the range of user choices, and/or leaving out application features that cannot be simplified. Although providers have taken some steps in all these directions, the extreme simplification strategy does not appear competitive (as of the early 1990s) in the evolution of PCs.

Table 4

Characteristics of Items Requiring Investment of User Effort: 1980s and Early 1990s

INVESTMENT ITEMS	LEVEL OF USE	EFFORT TO LEARN	OVERALL DESCRIPTOR
Equipment			
Connection	O/I	S	En
Switch settings	O/I	S	En
Start-up test	O/I	S	En
Operating procedures	F/B	S	En
Trouble indicators	O/I	S	Cr
Control and manipulation devices	F/B	S	Rou
Basic Software			
Operating system commands and results of commands	E	M	Maj
Start-up and access controls	F/B	S	Rou
Configuration controls and settings	O/I	S	En
Interface controls and settings	O/I	S	En
Security measures	O/I	S	Rou
Utility programs for management control	E	M	Maj
Utility programs for troubleshooting and damage control	O/I	S	Cr
System Organization			
Organization of storage (partitions, directories, files), and means for control	O/I	M	En
Access procedure	O/I	S	Imp
Start-up procedure	O/I	S	Imp
Documentation procedures and support	F/B	S	Rou
Back-up procedures and support	F/B	S	Rou
Sharing arrangements and other emergency plans and procedures	O/I	S	Cr
Each Application Program			
Capabilities, limitations, and general characteristics	O/I	M	En
Commands, execution options, and results	E	L	Maj
Installation	O/I	S	En
Transfer existing data and files	O/I	M	Imp
Set-up	O/I	S	En
Basic menu and dialog box entries, basic macros	O/I	M	En
Special start-up procedures	O/I	S	Imp
Special back-up measures	O/I	S	Imp
Special measures to protect nontransferable data and files	O/I	M	Imp
Personalization and Customization			
Batch programs and macros	O/I	S	Imp
Menus	O/I	S	Imp
System optimization for use pattern	O/I	S	Imp
Other customization	O/I	S	Imp
Growth Path			
Maintain awareness of new hardware and software	F/B	S	Imp

KEY **Level of Use:**

O/I = One-time or infrequent use
F/B = Frequent, mostly background use
E = Extensive use as a group, but individual programs will vary in levels of use

Effort to Learn:

S = Small
M = Moderate
L = Large

Overall Activity Descriptor:

En = Enabling activity
Imp = Improvement activity
Cr = Crisis activity
Rou = Routine activity
Maj = Major activity

performed and how to produce effective documentation for them. Finally, because most of the *Enabling* activities are both very visible and vital to the success of individual equipment and software providers, this class of items is probably the easiest for which to obtain outside assistance.[20]

A second set of items, also seldom used and often (but not always) associated with equipment or software changes, have as their purpose the improvement, rather than the enabling, of operations. Typical means to achieve improvement include measures to optimize equipment performance and to establish procedures that ease and increase the efficiency of some common user actions. Many *Improvement* activities are easily performed, and some are fairly simple in nature. However, others (for example, building complicated macros and optimizing system performance) can be quite complex and may require considerable detailed understanding of both computers and the particular software. These measures are seldom strictly necessary—as opposed to desirable—because their absence usually will not block quite intensive use of a PC by someone with only limited training and experience.

A third set of items includes a number of mostly physical, frequently used procedures that are rather easily learned and quickly become comfortable to most users; these are termed *Routine* activities. Maintenance of skills in performing most of these items is relatively simple because of their pattern of widespread use across all types of programs (and, increasingly, all types of computers).

A fourth set consists of a variety of *Crisis* control activities. The emergencies that call for their use can be of many types and can result from combinations of hardware, software, and human failures. Examples include: a need to recover a lost file, damage to a disk drive or diskette, a lock-up of the processing activity, recovery from a program "bug," and many

---

[20]A potential exception to many of these remarks arises in the transfer of existing files and data either to new equipment or (sometimes) in moving them to a new version of an application program. Usually no difficulties occur with the first upgrade of an application and its associated data, but over time (particularly if one or more versions are skipped) older files may become nontransferable—forcing storage on paper, or with the original (no longer used) application program, or some other special arrangement. This problem arises from the rapid pace of change and can complicate the use of a PC whenever long-term availability of data is important. The above remarks also apply to the transfer of macros.

others.[21] The quandary users face here is that there really are only a few of the emergency procedures worth learning fully in advance. Most of the procedures will seldom if ever be needed and, if they are, it may be sufficiently long since they were learned that their details will have been forgotten. What is much more important than simple application of learning time is that users apply a great deal of self-discipline in following a limited set of routine, anticipatory safety procedures (such as frequent file back-ups to multiple back-up diskettes, or equivalent). If this is combined with learning the key features of the available control and repair resources—that is, what the various emergency procedures can do, and where and how they can be accessed—the user will have done most of what is worthwhile. Beyond that, experience will be the main teacher.

The last category covers *Major* activities: learning to use effectively the appropriate functions of both the applications and other significant programs. This category focuses on achieving an ability to conduct all the activities whose successful accomplishment can justify the user's (or the organization's) decision to acquire a computer system, as well as the user's personal investments in learning to employ it. Many of the programs involved are likely to be large and complex and will receive heavy usage. Others are important for the user to be able to use, even if not frequently. As a result, these two groups of programs provide the focus for computer training courses and absorb most of the learning effort invested by all types of users.

Success in the learning process in the Major activities area is the *key prerequisite* to gaining the benefits that PCs can offer. Limited skills and knowledge concerning items of the first three categories of **Table 4** can result in an inability to work rapidly and in frustration at one's incomprehension of a feature or ignorance of how to do what one wants and knows is possible. Frustration can be a source of discouragement, and even fear. However, if limited assistance can be obtained, the lack of skills in these areas doesn't necessarily keep one from getting a lot of use out of a PC—which is just what inadequate capabilities in the last investment category *can* do.

---

[21]This list of crisis situations contains some that may be transient and thus offers only a "snapshot" of the early 1990s. In contrast, the self-discipline types of activity described described later in the paragraph may have very enduring aspects.

2.2.2 Usage Patterns

While the impacts of inadequate competence in the *Major* items of **Table 4** are very real, the requirements should not be overstated. Most applications have an enormous amount of code and many functions which most users are unlikely to need to know. The nature of the pattern of use is shown in **Figure 1**, which illustrates the variety of reasons why different functions of a program either are, or are not, used. The blocks in the diagram indicate different definitions; their sizes are *not related* to the sizes of instruction sets or computer code in the various categories.

The break-out and subdivisions shown for the "Used" segment illustrate a recursive aspect frequently encountered in the use of computers (and in many other information activities). Out of a collection of programs held by a user, normally only a few will be employed frequently, while others will have much lower usage rates—for reasons similar to some shown in **Figure 1**. A user who has acquired a collection of Shareware or Public Domain Software may have a

← ***Range of Application Functions*** →

Level of User's Understanding of Function	Functions understood; ready to use	Functions understood; ready to use	"Plan to learn," but not yet understood	Unaware of existence of function	Functions considered not worth learning	Functions irrelevant; no effort to learn
Perceived Value of Function	Not worth using	Valuable	Expected to be valuable	Possibly valuable	Not likely to be valuable	Not valuable
Status of Use	Not used	Used	Expect to use	Not used	Not used	Not used

One or few uses over time	Frequent but brief, simple uses	Extensive, steady uses

Figure 1

Use of Major Applications Functions

fair number of programs that, like much of the code in an individual program, never really get used!

Moving from the library level to individual programs, we come to the situation covered by **Figure 1**, which applies to both applications programs and support (operating and utility) packages. At a still more detailed level, within a single program many major functions (for example, print operations) offer a variety of options, some fairly specialized and complicated. Less trained users will view these functions somewhat as they view the program as a whole: they will tend to learn one or a few of the standard options and will ignore (and often remain unaware of the existence or the potential of) the rest. Thus, from the software library to the individual program function, there are strong similarities in the pattern of usage of the resources offered at each level.

2.2.3 Elements of Learning

Learning how to handle a computer system is itself a particular type of information package—one of the many possible information "service" packages. At the start of **Chapter One**, we described the operation of a PC as constituting a communications *process*, employing a variety of special languages (e.g., programs) to convey the *substance* that forms the computer's *processing* instructions and a conventional language, or set of languages (for input and output materials), to provide the *substance* that instructions *operate on*. Neglecting, for the moment, any consideration of format (which, for the learning situation examined in this section, would be the particular instructional format used), the roles of two of the three defining ingredients of the package "how-to-use-this-computer-system-to-run-that-set-of-applications" are established by this schema. Learning activity requires learning both *(i)* the needed and allowable "user control actions" (or processes), and *(ii)* that substance of the special languages that provides guidance on how, when, and in what combinations each action should be performed to achieve a particular feasible result.

Another way of describing the required computer learning is that it is broadly concerned with:

- acquiring a set of *skills*, that is, "learned powers of doing some things competently," and

- gaining a particular area of *knowledge*, which amounts to establishing "acquaintance with or understanding of" a body of information.

The first of the above establishes user capabilities in a set of action areas—it constitutes the human energetic input to the *Process* elements of the overall package. The second covers user mastery of relevant goals, means, limitations, and related subjects—that is, the needed human understanding of the *Substance* elements involved.

The above material implies a close connection between two human capabilities and two computer system elements. It suggests that Knowledge is directly related to Substance and Skills to Process. Thus, every computer activity or product in a given general format and every discrete step in learning how to conduct the activity or use the product involves a mix of Skills and Knowledge on the part of the user. Conceptually, therefore, each activity or step can be located on an axis that defines how much it represents a skill and how much a particular type of knowledge.

Figure 2 illustrates this concept as a means for understanding better the nature of different types of activities. It shows three sets of entries along the Skill-Knowledge axis. The first (**2-A**) is a high-level view, for the print-on-paper and successor electronic formats, which encompasses a selection of the general information-building activities involved and the products that result. The second (**2-B**) is narrower, covering only general computer operations (described in **section 2.2.1** and listed in **Figure 4**) for business or serious personal purposes. The last (**2-C**) summarizes the general relationships (already discussed) that underlie the features of first two diagrams.

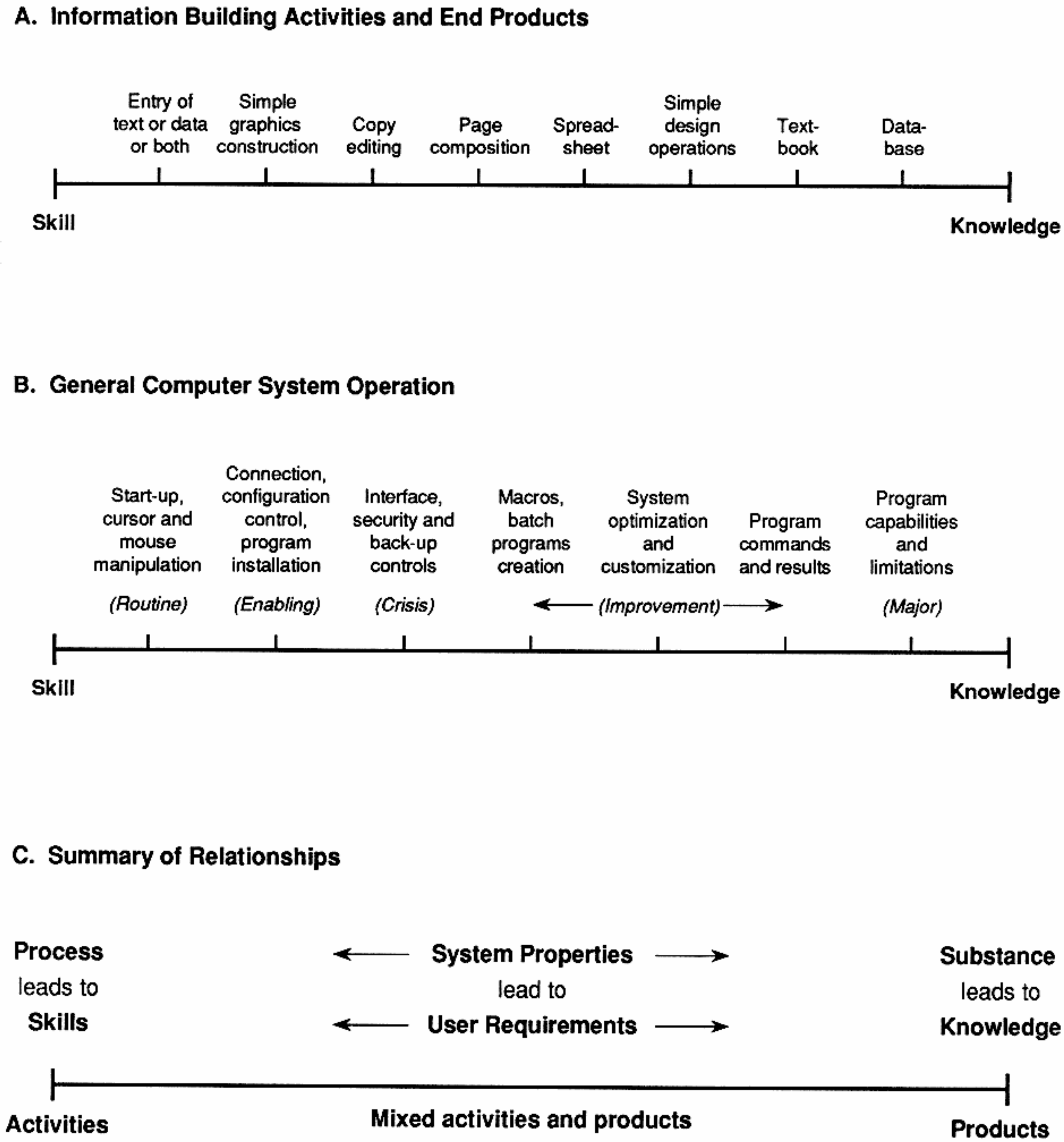
For the first plot, the entries move from "Text-Data Entry," at the extreme Skill end, to "Database" at the Knowledge end. The low substance content that characterizes certain text or data entry operations is demonstrated by the practice of U.S. companies that sent textual materials to foreign countries, like Korea, for keyboard conversion to electronic format. The entry operators in those countries did not know what they were typing—only that they should duplicate accurately the paper materials provided to them.[22] At the other end, at least some databases are relatively pure substance products, with only limited structure to support an uninitiated user who lacks the associated interpretive substance (e.g., knowledge) needed to understand and use the database.

The intermediate entries provide only an approximate ranking, since each entry can, by itself, cover a significant portion of the total axis. Text entry (of technical materials, for example) can benefit from considerable "local" substance knowledge. Graphics construction, editing, and design operations also can vary widely in complexity and in requirements for user substance knowledge. For any sets of specific examples, however, it should be possible to develop rankings that are fairly consistent both internally and across the sets.

In **2-B**, the end-points are a group of primarily physical, manipulative activities on the Skill side and descriptive material without any direct manipulative implications at the Knowledge end. The manipulative elements emphasize the physical, action-oriented aspects normally associated with skills, while the descriptive materials are essentially nonphysical but require associated substance for full understanding. Again, the intermediate entries are at best general rankings, and the assignments of position are entirely judgmental. It could be argued, for example, that the "Program Commands and Results" category is richer in substance than "Program Capabilities and Limitations" and in many respects it is. For example, it certainly is far more extensive and goes much deeper in its substance coverage, but it also is more self-contained and action-oriented. The key point is that it makes little sense to employ Program Commands and Results material only for reading or study (that is, as nearly pure Substance); the action orientation means that the material really comes to life only as part of an iterative sequence of substance acquisition followed by a period of application and *skill development*. Hence, the judgment that the shallower, more general material in this case has the purer substance content.

---

[22]A possibly apocryphal illustration of the danger of just a little knowledge is given by the report that when similar efforts were made in India the results were unsatisfactory. The reason given was that the entry operators, who knew some "true English," insisted on correcting spelling errors in the material they received—converting "labor" to "labour" and "aluminum" to "aluminium," etc.



Note: Italicized entries in parenthesis in 2-B refer to categories of overall activity descriptors in Table 4.

Figure 2

The Roles of Skills and Knowledge

In none of the plots of **Figure 2** is attention paid to the "size" of the learning (or performing) job associated with an entry; size is an entirely separate dimension for which crude estimates were given in the middle column of **Table 4**.

2.3 Acquiring Skills and Knowledge

We indicated earlier that, in the real world, there is no such thing as "pure" Substance/ Knowledge or "pure" Process/Skill. Even the most passive acquisition of knowledge requires some skill, if only to process and interpret written or oral symbols that convey the knowledge. Similarly, Skill has no meaning except in terms of accomplishing something, and this requires a knowledge of possibilities, purposes, and available means. There are, however, important differences in the ways skills and knowledge can be acquired. The ease of learning how to perform activities with different mixes of the two ingredients is subject to selective change whenever a new technology (such as computers) favors ease of learning one ingredient over another.

To illustrate, skills are action-oriented; they almost always involve specific, immediate goals; they require extensive (often continuing) practice; and they are usually learned by an iterative pattern that involves:

- gaining a modest (e.g., manageable) amount of knowledge through study, or by observation of and/or discourse with someone who already possesses the skill; followed by

- practicing the skill, first utilizing the knowledge already gained and then expanding on it by conducting "what-if" or trial-and-error activities; then

- recycling the above while also engaging in real-life operations (or competitions) as soon as the skills are adequate for practical use or display. Such activities will often involve another person or a group of people; skill acquisition frequently requires competition and/or cooperation.

In contrast, Knowledge is more contemplative—except when acquired to support a needed skill! Often its acquisition will involve long-term goals, rather than immediate ones. This is typical (in concept, at least) for much of formal education, although, again, there are strong exceptions—the most common being the short-term incentives that suddenly are aroused by the announcement of a pending examination! In many situations, the goals for knowledge acquisition may be defined only weakly, and there can be great uncertainty as to whether the specific knowledge being gained will ever be used in a purposeful manner. Any important knowledge-gaining activity will involve intense personal intellectual efforts.[23] Traditionally, these have been based primarily on absorption of printed alpha-numeric material, supplemented by discourse and argument with other individuals who have similar interests. More recently, electronically displayed materials and audio, graphic, image, video, and computer-interactive inputs have begun to be used; however, their overall impacts for purposeful knowledge acquisition were still small in the early 1990s.

The knowledge acquisition activity frequently takes place under the guidance of a teacher or an associate, but much of it also will be undertaken without significant outside assistance. Knowledge can be used or displayed—sometimes usefully and often boringly—and it can be developed in unstructured ways, as in social conversations; but it cannot be "practiced" in the sense a skill can. The closest one normally comes to acquiring knowledge by formal trial and error is the pleasant but often inefficient activity called "browsing"![24]

2.3.1 What the PC Can Do to Help

Enter the PC. What impacts has it had, or can it have, on the processes of learning or using information packages that incorporate very different mixes of skills and knowledge (including the packages labeled "the PC and its applications")? The essence of the answer to this question is the direct result of two key attributes of PCs in the *late 1980s and early '90s*:

- PCs are superb at performing large numbers of manipulations of many types on information contained in a wide variety of alpha-numerical-graphic formats as well in some acoustic and image-based formats.

- PCs are inadequate for extended careful reading and for a variety of reading-related information absorption activities. This will remain the case at least until a computer display becomes available that offers quality, size,

---

[23]This is not intended to ignore the more experiential and physical means for gaining knowledge, such as laboratory or field work; these activities are important and often critical. Much of this type of work depends on physical skills, but the analytical component normally will be primarily intellectual.

[24]*Hypertext* provides an approach for computer-aided browsing that may turn out to quite efficient, but this still is uncertain. By the early 1990s, materials in hypertext form were available and useful for work in some specialized fields (such as law) that already made extensive use of citations and references. However, it is liable to be quite a while before substantial amounts of more general browsing material become available.

portability, and other comfort features comparable to those provided by paper-text products.

At the start of the 1990s, the operations of PCs provided a far better environment for skill-intensive activities than for most types of knowledge acquisition endeavors. However, this statement must be interpreted with care: hidden in many substance absorption activities are critical skill-intensive components, such as searching for, accessing, extracting, organizing, moving, and otherwise operating on information to support current (or potential future) knowledge-acquisition efforts. Also, some of the current PC weaknesses related to substance absorption are transient and subject to change as display technology advances or as greater use is made of certain special capabilities of computers.

The most obvious of these are the transparency/simulation capabilities made practical by the computer's speed and flexibility, described in **Chapter One** in connection with their use for building dynamic Graphical User Interfaces. GUIs provide an excellent demonstration of ways whereby operations can be simplified and made "friendly" to less skilled users. Another example is the growing use of WYSIWYG ("what you see is what you get") screen displays. Before these became available, the use of arbitrary, definitional symbols at appropriate points in material was the only way to indicate many features of the eventual printed output. In contrast, WYSIWYG displays show on the computer screen almost exactly how the printed output will appear.

In effect, applications of the transparency/simulation capabilities of computers have made at least three contributions to easing the process of learning to use applications effectively:

- They have converted a portion of the set of arbitrary (thus hard to learn) symbols used for instructions in computer application programs into a group of logical, intuitive manipulations of familiar, representational screen images.

- They have turned part of the needed knowledge-acquisition process into skill-practice operations by enabling and encouraging a great deal of experimentation and "trial-and-error" testing by users. Experimentation leads to both direct gains in knowledge and the construction of user perspectives and frameworks that can ease later acquisition efforts. The overall process is very natural, and it stimulates the learning activity and makes it more enjoyable.

- They have achieved (as noted in **section 1.3**) considerable de facto standardization through the practical requirement that all applications (for the first popular GUI systems) be written to operate with a single interface specification. While this may change as new GUIs and more advanced systems are introduced, the residue of early standardization is unlikely to disappear rapidly.

These means for simplifying the process of learning to use computer applications (*and to acquire many other skills*) can be summarized as: movement from use of arbitrary, symbolic command and display structures to use of more logical and intuitive, imagery-based methods; conversion of part of the learning activity from means that require intensive knowledge acquisition activities to more action-oriented, experiential approaches; and increased standardization of the details of many individual computer operations. These capabilities, especially in combination, are unique to computers. Because they demand great processing speed, they became practical only after PCs reached the performance level of the late 1980s.

GUIs and WYSIWYG are not the only examples of techniques that exploit the computer's speed and flexibility in ways that may help users acquire desired skills and knowledge. Whole families of new formats are arising, with features simply not feasible prior to the development of relatively powerful computers, some of which may be very important to future learning activities. Examples include *Hypertext*, CD (Compact Disk) Interactive systems, Multi-media,[25] and various combinations of these formats. All offer tantalizing possibilities and are receiving attention, but, as of the early 1990s, none has achieved significant general market penetration. For this reason, their practical potentials for widespread use are not yet established. Further, all of them face barriers raised by near-term costs for quality program production (which are likely to be high) and the consequent difficulty in rapidly building the kinds of markets needed for economic viability.

Other more conventional means already have been used widely to assist the process of learning new computer applications programs. Some (such as complexity-hiding) depend

---

[25]Multimedia is an ill-defined term. At its simplest, it involves a product that has different portions prepared in different modes (written text, voice, images, etc.) but without any implication that the materials in all these modes are broadly manipulable by computer. A more advanced version of Multimedia might require that all the materials employed, regardless of mode, can be "read" and manipulated in significant ways using the machine.

strongly on continuing advances in technology. Others (such as improving program architectures and designs, responding carefully and fully to user feedback concerning problems and desires, and gradual standardization in response to market pressures) are more independent of technical progress. No matter how far any of the approaches is pushed, there are limits to how much the learning burden can be eased—especially because opportunities for increased functionality, at the cost of greater inherent complexity, are open-ended.

Another reason for the limits is that the major application packages make two important (closely related) contributions which have received only limited attention here: (*i*) significant fundamental quality improvement in work output, and (*ii*) user training in what often will be totally new skills.

Quality improvement shows up in such ways as: lower output error rates; more thorough supporting analyses; more attractive output appearance; faster, more timely and more complete work performance (provided these attributes are not bled-off in an effort to obtain "super-quality" in other areas, like more detail, more elaborate appearance, etc.!); improved communicability; and many others. To achieve some of the desired results, a user must learn a new set of techniques, such as how to make a set of graphs more comprehensible or a page more attractive. Until these noncomputer skills are somewhat developed, the related quality-improving operations of the computer can't be used effectively.

The second area, that of more general skill development, has an enormous range of possible requirements. The minimum end is similar to that described for exploiting potential quality improvements—the non-computer aspects of the basic skills an application is intended to support (for page composition, spreadsheet operations, preparing drawings of various types, etc.) may need to be mastered before or in conjunction with the associated computer skills. In many cases, the need for acquiring these basic skills will be job- or task-related and largely independent of the fact that a computer will be the means for expressing the new abilities. The learning process probably would require even more user effort if no computer were involved!

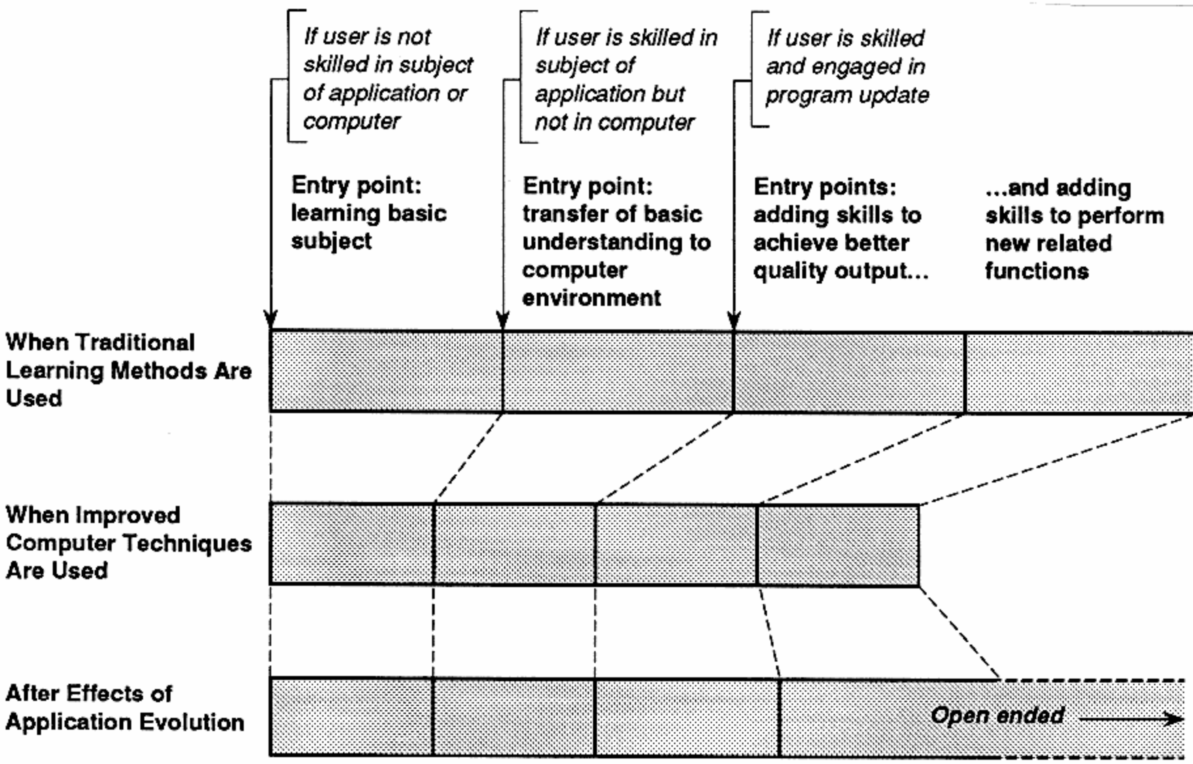
Beyond the minimal case is an open-ended situation. Extensions and related functions that appear "natural" to add to existing major applications can arouse great interest on the part of designers and marketers. If developed and offered, some of the new capabilities—requiring new learning—almost always will prove attractive to some or most users. This feature applies very broadly when PCs are involved—to hobbies and recreational activities as well as to work-like applications. In making decisions, the user investment effort "charged" against the PC should reflect the fact that much of the total effort may apply to acquiring a wide variety of new abilities not in themselves concerned with computers except as vehicles for their application. As happens when people use more conventional methods to learn new subjects, the results of the efforts have the potential to improve and enrich their lives in a variety of ways.

2.3.2 The User Balance Sheet

The general user learning situation is illustrated in **Figure 3**. To get started, a new user may need to learn the basic subjects which are the concern of the application being mastered. The next (and sometimes almost simultaneous) step for the user is to transfer this understanding to the computer environment; this involves learning the vocabulary and the grammar of the application program's command language. Users already experienced in the subjects that the application supports but who have never used computers or the particular program involved also go through this second step. Thereafter, both types of users must acquire additional subject and computer skills if the potential quality and full range of functionality of the application are to be exploited well. Part or all of the process must be repeated whenever a new program (or version of a program), offering new or different capabilities is installed.

Most of these processes can be conducted with less effort if the designers of the computer, and of the application program in question, have taken advantage of the techniques for simplifying the learning process discussed in the previous section. The general background of users and their skills in activities important to PC operation (such as typing) will also affect the learning process. Experienced users can be expected to learn faster than inexperienced ones, even if their experience was gained working with applications other than the one under study.

As an application evolves over time, continuous interplay will occur between improvements that ease a new user's learning efforts and increases in program quality features and functionality that require additional skills as the price of the new benefits. A likely near-term pattern for individual major applications is that more potential learning requirements will be added, by growth in available quality and functionality (of increasingly marginal value to



Note: Lengths of bars and bar segments are indicative only and are not intended as actual measures of actual effort.

Figure 3

The Learning Process:
Areas to Which Effort May be Devoted

many users!), than will be eliminated by learning simplification. However, from the point of view of the user population as a whole, there need be no clear loser to this interplay. Some users will seek the lowest practical level in the learning effort to which they must commit, while others will revel in acquiring and then using new skills. To a considerable extent, both views can be satisfied—although this partly depends on how much businesses or other outside sources require users to employ more of the full range of available capabilities than the low-commitment users would prefer.

Where does this leave the users? What price must they pay to become effective recipients of the benefits computering has to offer? Is there a realistic minimum price? a desirable level of personal investment? Clearly there are few general answers to these questions, if only because users come in so many varieties. There are, however, some general trends.

First, it is clear that, at the minimum skill/use level, PCs are steadily becoming easier for new users to employ. The common concepts of "computer literacy" during the early 1980s implied that users needed to acquire some ability to program, usually in BASIC. Few new users in the early 1990s are likely to regard this as a requirement. If computer display improvements enable production of effective electronic "reading devices," different types of highly useful personal "computer" systems (with associated applications software) may, in the future, range in complexity almost continuously—from those so simple that essentially no learning is needed to extremely sophisticated ones appropriate only for the highly skilled.[26]

Given this spectrum of possibilities, users will try to distribute themselves (or be distributed by their employers—or the employers' suppliers!) at positions where they feel that the values they receive from PCs are in good balance with the efforts and financial costs they must devote to acquiring, learning, and maintaining hardware and software skills. The values and the costs will vary over time. The dynamics of user learning costs were described earlier in this report; the change pattern for user benefit values is likely to be equally complicated.

The benefits of using a PC will depend in part on extensions of what can be done with computers (derived from a combination of innovations in application concepts and advances in hardware and software) and in part on a variety of business, government, and social pressures. As with other types of new technical applications, these pressures will not always be clearly beneficial to society, even when they increase the value of computer skills. By their nature, computers encourage the development of greater detail in records, finer quality in presentation, and greater breadth in output distribution, to an extent that easily can exceed the levels appropriate to a situation. They are a bureaucrat's dream—with results already visible

---

[26]This area is discussed in the reference in note 14. The spread of types of "computer" systems described here is another reflection of the trade-offs between extreme simplicity for users or more complexity for the sake of increased capability/functionality (as discussed in note 9).

in a number of business and government activities! However, whether desirable or not, the pressures on the individual users can appear very real.

In many occupations, the value of gaining a specified depth of understanding of an application (or set of applications) will be established at a detailed and unambiguous level by the work organization. In more flexible jobs, and especially for personal uses, the appropriate level is apt to be a moving target, influenced at times by various forms of personal competition (primarily at work) but set mainly by the capabilities the computer is offering at the moment of user decision. The particular capabilities of value to a user will depend both on those available in a given application which the user wants to exploit and on the user's decision as to which applications of what types should be subjected to this determination. This selection may be confined by the user to a narrow sector of functional activities or expanded to a wide variety of different fields of interest.

To summarize the situation of these users:

- For major applications, the learning effort to master "bare-bones" functions will decrease, continuing the past trend of making capabilities more available to inexperienced and unskilled users.

- For the same applications, the full range of capabilities can be expected to grow, as in the past, to a level where probably only those with semi-professional skills (or better!) and interests may or will be able to employ most or all of their features.

- The number and variety of (both major and minor) new applications of interest also should exhibit strong growth; all indications suggest that the opportunities for new computer applications are effectively unlimited.

Many of the new applications may be variants on or significant extensions of existing programs, but some will be radically new. In the past, the applications that stimulated market penetration of PCs were concerned with supporting (better and faster) a series of activities already common. This meant that for each application there were past analogs to the computer-based system. There also were existing sets of user skills; these could help the application get started but later might act as sources of resistance to some lines of possible improvement. In contrast, many of the more important applications of the future will probably employ formats not possible prior to the development of high-speed, high-capacity PCs, and/or be concerned with supporting processes for which (for the same reason) there are no past analogs and fewer, if any, well-developed sets of definable user skills.

If this vision is substantially correct, then the availability of growing numbers of new applications will eventually lead to essentially unlimited opportunities and requirements to learn new skills useful to operate and control PC programs of business or personal interest. While some of these skills may be familiar and already in widespread use, others may be so different and new that they are still not formally recognized and may not even exist yet either in practice or imagination.

In this context, answers to questions concerning how much users should invest in understanding computer applications become almost indistinguishable from the same questions asked about their investments in personal education. If the display qualities needed to compete with paper products (for reading and absorbing information) are added to the PC's other growing capabilities, over time personal computers can become the primary way in which literacy—in its broadest sense, incorporating traditional linguistic aspects and also many related artistic and musical elements—can be expressed. This will make the PC (or its successors) a key participant in all but the most elementary formal learning processes.[27] As a result, user choices of how much to invest in learning skills that help him or her exploit what the computer can offer will be integrated, in a very fundamental way, with his or her choices about the personal education level to be sought. In turn, the freedom of choice available, and even the extent of understanding of what these choices are, will depend on the social investments that have been made in what is taught in our schools. In this manner, the computer-related choices can evolve to join the central decisions of society and of an individual's life.

---

[27]This remark *does not* imply that the computer will be acting as a "teaching machine" in any but the narrowest sense. It will provide a display device that offers some advantages, relative to print on paper, for reading, absorbing, and using information. It will provide a highly flexible tool for user experimentation of many types. And it can aid learning in a number of other ways (some not yet discovered). In almost all fields of learning endeavor, it will be a participant and an aid—but not necessarily a substitute for a teacher.