Title
A Model for a Seamless User Environment

Author
Stevens, D.F.

Publication Date
1989-07-01

LBL-27533

Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA, BERKELEY

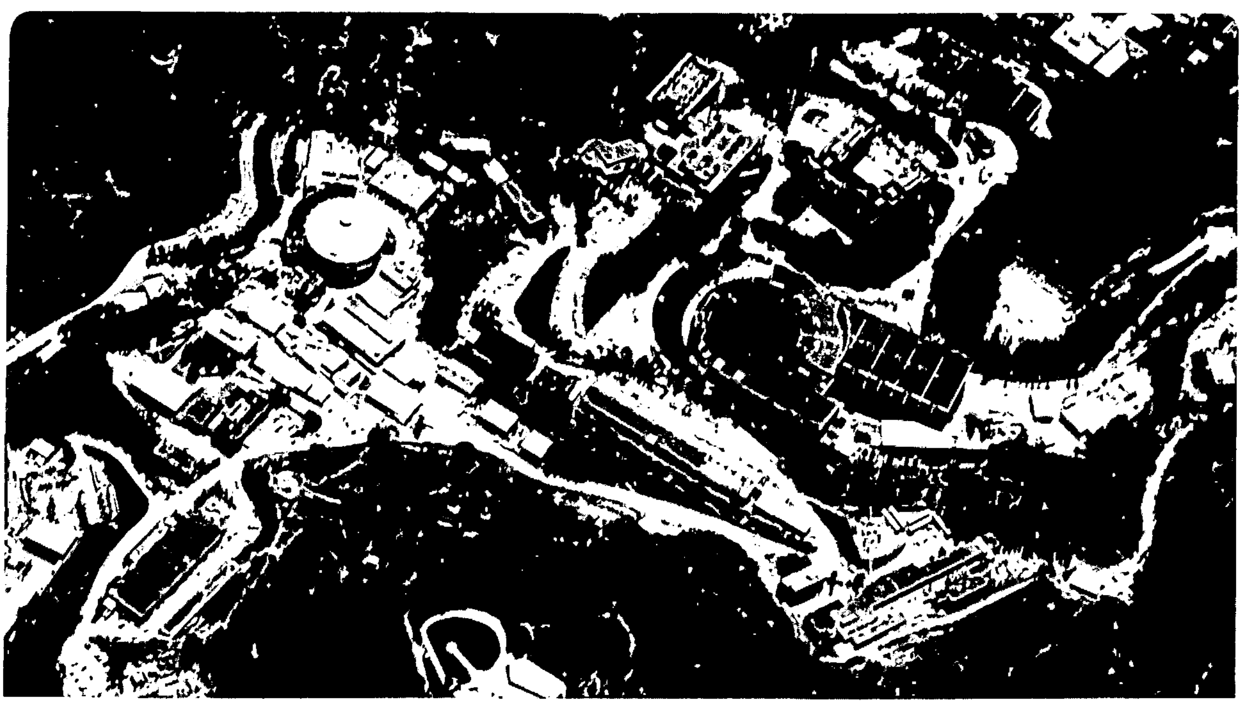
Information and Computing Sciences Division

To be presented at the Beall IT Forum '89,
London, England, October 30–November 3, 1989

A Model for a Seamless User Environment

D.F. Stevens

July 1989



Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098.

LBL-27533

A Model for a Seamless User Environment*

D. F. Stevens
Information and Computing Sciences Division
Lawrence Berkeley Laboratory
1 Cyclotron Road
Berkeley, California 94720

Preface

The *seamless user environment* appears to be an idea whose time has come. It has long been a dream of information technology visionaries, but only recently have we begun to see the emergence of the infrastructure and products necessary to the realization of that dream. Work is progressing on several aspects of an overall seamless environment, but there is at present no generally-accepted architectural framework around which to structure a discussion about what such an entity might be. As is often the case when evocative vocabulary is applied to complex realities or (as in this case) potentialities, it is rare to find two interpretations that are more than approximately equivalent. The model described in this note is intended to provide a foundation for one possible interpretation.

Before moving to the user *environment*, however, let us look briefly at the user-system *interface*, wherein lie the orgins of many of the seams we wish to eliminate or conceal.

The User-System Interface

The user-system interface available through a modern workstation is a complex combination of many disparate elements, including:

- three different kinds of *objects* (namely, symbols, physical devices, and system internal constructs);
- the *actions and operations*, both actual and symbolic, permitted on and with the various objects;
- the *interpretations* applied to objects and actions; and
- the *vocabulary* used to define and describe them.

The user interacts directly with only two of the object classes—physical devices and symbols—and his direct interaction with symbols is limited to observation and interpretation, inasmuch as manipulation of the symbols must be accomplished via manipulation of the physical devices. (See Figure 1.)

* This work was supported by the U. S. Department of Energy under contract No. DE-AC03-76SF00098.

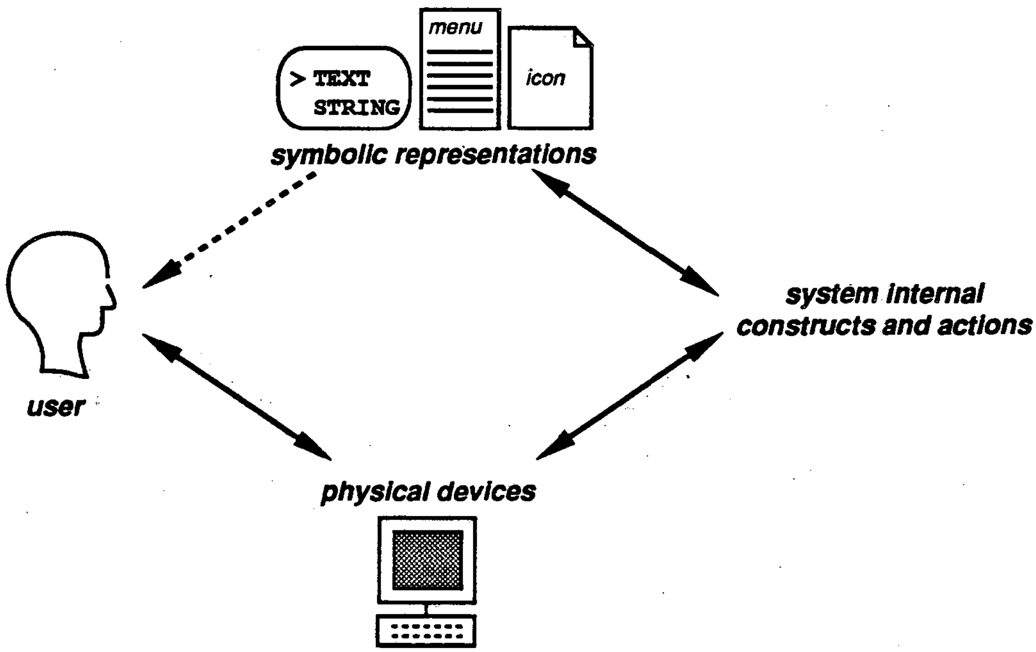


Figure 1: The User-System Interface

This extensive and diverse collection of physical and logical entities leads to an interface that, while it may be "easy to learn" or "user friendly" or extremely desirable in some other fashion, is almost certainly not straightforward. There are numerous reasons for this, including the following:

- The repertoire of operations and actions available ranges from very low level (tilting a VDU screen; executing a bit-level machine instruction) to very high level (rotating a molecule in space; paginating an entire document, with automatic placement of footnotes, (re)numbering of pages, and update of the index, all as a single operation).
- The (system-oriented) complexity of an operation cannot generally be predicted from the (user-oriented) complexity of the action required to perform it. Simple operations may require relatively complex sequences of actions (deleting a document in a protective icon-based system can involve selecting the document with a mouse, pressing a function key or selecting a menu option, and reconfirming the intent), while some extremely complex actions can be accomplished rather simply (the pagination example given above can be invoked by the selection of a single menu option).
- The set of symbols available can be limited to the graphics that represent characters input through a keyboard, or can be expanded to include pictorial objects of arbitrary complexity.
- Many objects, symbols, and actions have context-sensitive interpretations.

Effective use of the system demands that the user maintain familiarity and competence with all the objects and actions involved. He must learn not only how to deal individually with fundamentally different kinds of objects, but also how they interact with each other. The specification and accomplishment of most tasks proceeds in a manner similar to the following:

1: The user translates his real task (the printing of a document, for instance) into the actions he must perform. This may either be done as a whole, before he begins the actual operation, or as the operation progresses, with each step informing its successors. It requires an understanding of the *capabilities* of the system at the application level, but does not necessarily require any knowledge of the inner *mechanisms* of the computational procedures that create the application.

2: The user performs the necessary actions. In a command-driven system, this means entering a command; in a menu-driven system, selecting the right choice (or sequence of choices) in a menu; in a window-and-icon system, selecting an icon and manipulating it. In all cases, the user manipulates one or more physical devices in ways that are understood by the system.

3: The system interprets the changed state of the physical devices manipulated by the user, updates the screen-resident representation of the system, and interprets the changed state of the system universe as now represented by the screen. The user interprets the changing display on the VDU to monitor his (and the system's) progress.

4: From time to time, the system decides that it has enough information to change the system itself to conform to the instructions communicated by the actions of the user, or implied by the changes in the screen representation. These changes to the system are communicated to the user, occasionally through changes in the physical devices, but more usually through changes in the location and/or appearance of the symbols on the screen.

5: When enough information has been transmitted from the user to the system, the system and its devices perform the task requested.

This scenario makes it clear that there are (at least) four views of any task:

What the user *actually* does; i.e., what buttons he pushes, etc.

How the user *describes* what he does; how he talks about it, to himself or to others (turning up the gain; opening a window; going through a door; looking at the other side of an object; etc.).

What the *system actually does* (creating and maintaining tables, compiling and/or executing programs, defining and modifying files, etc.)

What is *accomplished*; e.g., the document printed or simulation performed.

The second of these viewpoints—how the user describes what he does—defines the user environment. It contains the objects that populate the user's universe of discourse and the user-level operations that are permitted therein. It is between, and within, user environments that we find the seams that cause us so much trouble.

Of Environments and Seams

To speak meaningfully of an environment without seams we must first have some understanding of what constitutes a seam. We shall take as primary definition that *a seam is a visible join*, a border that requires conscious effort to cross. The essential defining characteristic of a seam is not its "bordership", nor the magnitude of the difference of the realms on the two sides of the border, but the *visibility* of the border, the necessity for the user to become aware of its existence.

As noted above, seams can exist both within and between environments. The latter case is generally more obvious, as, for example, when one moves from a command-oriented program-development environment to an object-oriented simulation environment, or from a text environment to a graphics environment. But even within a single environment it is possible to encounter many seams. For one thing, the system designer may have created multiple *de facto* environments where the user sees only one. A familiar example is document creation, where the user's single document-based environment is partitioned by many systems into half a dozen specific environments (e.g., text entry, formatting, tables, pictures, spreadsheets, and printing). Another classic example is the one-time engineering calculation. In early systems, the user could not just state his problem and receive the answer; he had to be painfully aware of such *system* activities as program entry, compilation, linking, and loading, as well as program execution, and he generally had to request several of those incidental tasks explicitly. In modern systems, many of these system-induced decompositions are hidden from the user, so that he need not be aware of the necessity to "open" files or to allocate space, for example. In other cases, to get his work done, the user must exhibit knowledge of network protocols, routing, login procedures at multiple sites, multiple text editors, and so forth. Each of these system-oriented tasks or structures is a seam; each time the necessity to deal explicitly with one of them is removed, a seam is eliminated.

A different sort of seam shows up when the coupling between control devices and the objects they control is insufficiently close. If there are delays, so that the user must wait for the system to "catch up", or if system motion is visibly quantified when the user wants continuous motion, the user is forced into an awareness of the actual system processes that underlie his actions; in each of these cases a seam is introduced.

As a final example of seams, we can consider the difficulties often encountered in seeing (literally) the results of one's efforts. Professional programmers began to encounter 'wysiwyg' (what-you-see-is-what-you-get) text editors some years ago, but it took Visicalc and its successors to bring the advantages of instantaneous results into the domain of the ordinary user. This revolution continues today in the recent emphasis on powerful 'visualization' systems.

After one becomes accustomed to a well-designed application, it requires conscious thought to separate the real actions of the user (movement of the mouse, for instance) from the effects produced on the screen (movement of the pointer with which the mouse is coupled). The user is free to think in terms of the metaphor of his environment, and no longer needs to be aware of what the system is really doing; a measure of seamlessness has been achieved within this limited environment. In a fully seamless environment all of the user's work would be conducted in terms of appropriate work-oriented metaphors, shifts between metaphors would be automatic, and the user would never have to enter a system-oriented mode of operation.

The Seamless Environment Defined

The short definition of the *seamless environment* is "The world-computer, available at the user's workstation." A more complete definition requires some exegesis on the terms "world-computer" and "available at".

The *world-computer* is the collection of all the computing resources the user may need, wherever they may be located. In some contexts it might be desirable to limit the discussion to "all the computing resources *to which the user has authorized access*, wherever they may be located," but we may consider authorization to be just another seam. (In a truly seamless environment, a user would have authorized access to all the resources he needs.) There thus needs to be a connection of some sort between the workstation and the rest of the desired resources. While this sounds like a significant barrier, there is a great deal of anecdotal evidence suggesting that all unclassified networks are in fact already interconnected through unofficial bridges and gateways (even though in theory and officially they are independent). For those who have the necessary special knowledge, then, an embryonic and underpowered form of the world-computer already exists. (The requirement for special knowledge means that the current approximation to the world-computer is rather seamy, however.)

We should emphasize that the word *computer* does not appear in the definition of the world-computer; *computing resource* is used instead. This is not a mere stylistic device to avoid repetitive terminology, but an explicit reminder that computers, *per se*, are not the only computing resource to which a user might wish access. There are programs and data and graphics out there, too, that are likely to be of some interest, and the seamless environment requires seamless access to all of them, as well.

The world-computer provides a user with a complete environment, but it is the *availability*—in an extended sense—that makes it seamless. There are degrees of availability. Fossil water is *available*, from the interior of glaciers or the depths of extraordinarily deep lakes, but it is not so available as tap water. Electricity is *available* pretty much around the world, but most record players require conversion devices to achieve correct operation in all countries. The sort of availability that is necessary for a seamless environment is "available *as if it were on the user's workstation*." In the seamless environment:

the user can access any data anywhere as if it were on his workstation;

the user can read any file on any system as if it were on his workstation;

the user can run any analysis program he needs as if it were on his workstation;

the user can use any computer he needs as if it were his workstation;

the user can manipulate any visual image as if it were stored on his workstation;

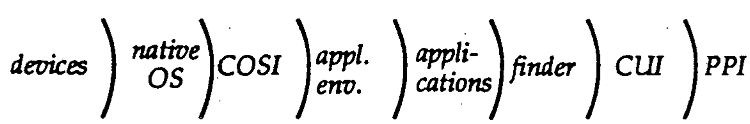
and so forth.

In the seamless environment, the user has apparently direct access to any desired resource; use of any resource conforms to his expectations; and there are no explicit translations necessary between the language of the resource and the language of the environment within which he chooses to work. In other words, he addresses the data he wants, not the database, the DBMS, the computer, or the actual storage device. He addresses the algorithm he wants, not the program, the language, or the compiler. He uses the language and objects of *his* chosen metaphor to accomplish the work he needs to do. He can display the results in the form he wants. Furthermore, all the applications he needs are *interoperable;* i.e., any work product of one step can be used as input for another step, without (explicit) conversion.

It is worth emphasizing that the word *consistent* does not appear anywhere in the definition or examples of seamlessness. Consistency is not a prerequisite, and is in fact most likely a detriment to seamlessness. (Remember Emerson.†) Stylus, mouse, and keyboard are surely not 'consistent' forms of input, but a user may wish to draw with the stylus, select objects with the mouse, and enter text with the keyboard. To be seamless, the environment must be *appropriate* rather than foolishly *consistent;* appropriateness is in the eye of the user, and changes with the user's experience and objectives.

Approaching seamlessness

The essential infrastructure for a seamless environment consists of *connectivity, bandwidth,* and *conversion.* Connectivity and bandwidth are being approached through the worldwide telecommunications network supplemented by implementations of the OSI model; their function is to provide pathways of sufficient capacity between the actual residence of the desired resources and the user's workstation. Conversion allows workstation commands to be understood at the various actual resource locations, and allows the images and responses generated at those locations to be displayed in meaningful fashion on the workstation. Needless to say, many, many conversion processes will be necessary to achieve anything approaching true seamlessness. But we can specify some generic levels that may allow us to reduce the problem to manageable size. We begin with an eight-layer onion model‡ :



devices: The physical devices of the world-computer, including computers, firmware, and microcode as well as storage and I/O devices.

† A foolish consistency is the hobgoblin of little minds.
‡ The layers of this model are not intended to correlate in any way with those of the OSI model.

native OS: The native operating systems that drive the devices.

COSI (common operating system interface): A "shell" that converts a native operating system into a standard virtual operating. POSIX is the first serious attempt at a COSI.

application environment: A sort of "software bus" that mediates between specific applications and the COSI. The initial attempts at seamlessness will take place within single application environments.

applications: The specific applications that make up a given application environment. It should be emphasized that 'equivalent' applications are completely interchangeable; a user who wishes to compare the results of two economic models need not learn two interfaces, for instance; he just tells the system to use Model-2 instead of Model-1.

application finder: A directory and linkage service that connects the desired application at the workstation with its counterparts, complements, and necessary resources, wherever located.

CUI (common user interface): A prescribed standard set of objects and actions for controlling computer processes. CUI serves the same function between applications and PPI as COSI does between native operating systems and application environments. A subset of CUI is the object of a current ISO standardization effort.

PPI (portable personal interface): A conversion/translation layer that converts the standard CUI into an environment with which the user is comfortable.

It is not necessarily the case that all eight layers are involved in every transaction. Once the connection between the workstation application and world-computer application has been made, for example, the *finder* need no longer be involved. It also should be emphasized that when seamlessness is achieved, all of the layers save the outermost will be hidden. To the user, the universe will be simply:

$$\left. \textit{world-computer} \right) \textit{PPI}$$

The rôle of the Infromation Technology Department — I: Attitude

It might seem that the achievement of the seamless environment would diminish the rôle of the Information Technology Department (IT), and that therefore it is not in the interest of IT to foster its development. While it is clearly the case that growing seamlessness will *change* the rôle of IT, there is no *a priori* reason why that change should necessarily involve a lessening of the centrality of IT in many modern enterprises. The actual effect will most likely be to some extent a consequence of the manner in which the corporation, and its IT staff, approach seamlessness.

The corporate IT staff can take any of several attitudes towards the development of the seamless environment. The most obvious are *enabler, observer*, or *opposer*, but there are also *oracle, unbeliever, dictator*, and *repaiman*. As *observers*, IT simply watch what is going around them, offering neither support nor opposition. An *oracle* is an observer who cannot refrain from predictive comments, but whose inherent ambivalence towards the situation renders all comments obscurely ambiguous. An *unbeliever* is an observer with a sharp eye towards the impossibilities and improbabilities of the situation, or, equivalently, an oracle who follows Cassandra rather than the Pythoness. The largest potential drawback with any of these rather aloof attitudes is that the users may find the presence of a central IT facility or staff to be *prima facie* evidence of a seam.

Another type of observer is the *repairman*. In this rôle, IT provide no assistance in advance, but stand ready to correct the mistakes of the "amateurs" as they charge into the seamless universe. This attitude is frequently combined with that of unbeliever, and compounds the negative features of the latter with the implied arrogance of the ignored expert; the combination exacerbates the natural inclination of the amateurs to conceal mistakes until disaster is upon them. A likely consequence is a (proper) insistence on the part of the rest of the corporation that *prevention* would have been preferable to *cure*, accompanied by an attempt to switch repairmen.

As *opposers*, IT fight the onslaught on several fronts: They point to the 'waste' of computing resources implicit in the many conversions and translations necessary to seamlessness; they emphasize the pitfalls of distributed data management rather than the user convenience; they tell horror stories about the spread of viruses and other vermin attendant upon global interconnection; they provide unwilling and inadequate support. Much of what they say is true, for seamlessness is not to be achieved trivially or carelessly, but their opposition is unlikely to withstand the thrust towards seamlessness already apparent in the more competent workstation systems. As users begin to see the possibilities they resent ever more strongly the barriers, and opposers are likely to end as victims, overwhelmed by the backlash that develops as the users achieve independence.

As *dictators*, IT embrace the concept of seamlessness, but force it to happen on their terms. This well may work for a considerable length of time, but unless they are extremely wise dictators, and steer the corporate environment into the paths the users would have chosen for themselves, there is likely to be an eventual revolt.

As *enablers*, IT recognize the inevitability of the seamless environment (or some approximation thereunto), and expend every effort to see that it is done right, and that the corporation as a whole benefits from it. It is my belief that the central IT function is a service function, and that its primary responsibility is to see that the power of information technology is constructively applied to the affairs of the corporation. This responsibility is met by enabling users to employ information technology in ways that are natural to their work, not by requiring them to develop an intimate knowledge of the inner workings of computers and networks.

It is also in the enlightened self-interest of IT to adopt the rôle of enabler, for at least three reasons. One is simple political expediency; the users who benefit may be persons of considerable corporate influence. The second is that, according to current experience, the more effectively the end users compute, the more the central computing load will grow; as you adjust your service offerings to meet the new demands of the seamless environment, including convenient, high-bandwidth access to the central facility, you will find significantly increased demand. After all, demand is clearly a function of the number of 'programmers' feeding applications into the system; in promoting seamless computing you increase the potential number of programmers many-fold.

The third reason to become an enabler is the education about new uses that your staff will gain in the training process. The introduction of revolutionary technology changes society, and seamless computing will qualify as a revolutionary technology. In the small, that means it will change the way your users do business, as they shift from merely speeding up the old processes to wholly new ways of working. These new ways will provide new demands for central IT services, if IT is perceived as a partner and is poised to offer support. One of the side effects of a well-administered enabling policy is that IT will be so poised and so perceived.

One of the most effective ways to become an enabler of this new technology is to become the end users' teachers and trainers. *Someone* has to teach us how to live in a seamless environment, and it might as well be IT. The new environment will bring a certain number of pitfalls as well as enormous freedom. Some of them, such as the problems of distributed data management alluded to above, are primarily technical, and will be solved (or not) by the professional informaticians. Others, however, are problems of behavior that apply to all users. For as seamlessness frees us from the restraints and constraints of current systems, it will also allow us to extend the effects of foolish or misdirected actions far beyond our own workstations and files. This is not the place to do any more than list some of the possibilities, as food for thought:

- Electronic junk mail: Not everyone shares our enthusiasms, concerns, or sense of the ridiculous; distribution lists should be created and used with a great deal of care.
- Epidemiology: A little bit of corruption can spread a long way in a short time in a globally-connected world; a 'harmless' prank can become a devastating catastrophe if it gets out of hand.
- Capacity planning: It is already hard enough; it will become still more mysterious when the whole corporation is generating applications, and running them at remote locations.
- Tool selection: Just because a Cray is available in the next county doesn't mean that it is the best tool for the job.
- Control: Whole new philosophies of enforceable control will have to be created and implemented.

The rôle of the Infromation Technology Department — II: The evolution of IT

The removal of seams may involve a lessening of the *visibility* of IT, but it is more likely to be accompanied by a broadening of the rôle than a narrowing. In fact, it may be that this broadening of our understanding of the functions that constitute IT is in truth the driving force behind the seamless revolution. As the term has been understood for the past few decades, *information technology* was strongly dominated by the idea of computer technology. As computing technology has become more deeply ingrained into the general societal infrastructure, we are finding it ever more difficult to separate the computing element out of other information technologies, such as communications, telephony, publishing, and the library.

The growth of networking, from the simple attachment of dumb terminals all the way up to the global web of interconnections necessary to the creation of the full-fledged world-computer, has made us all aware of the increasing interdependence of computing and communications. Computing may never attain the degree of intimacy with the other three information technologies mentioned above that it has already achieved with communications, but their dependence upon computing is growing apace. The digital PBX is bringing computing into telephone services in a fundamental way; computer-based document processing systems and desktop publishing are not only changing the nature of publication services, but are promising—or threatening—an 'end-user' revolution in publishing as profound as the one we are seeing in computing; and computer-based catalogs, reference services, and CD/ROM reference digests are starting to bring significant computer-based changes to library services, as well.

To continue to provide a suitable corporate infrastructure for information technology, the formal IT function must extend its breadth and vision to encompass this new breadth of computer-based applications.

Organizational implications

So far we have considered seamlessness only from the point of view of the individual user. It will also have a considerable impact on the corporation as a whole, not just in the installation and application of the technology, but also in more abstract organizational and sociological ways. The sociological impact encompasses the changes in personal relationships that will accompany the introduction of new technology into each corporate environment; the organizational impact is embodied in the structural changes that follow the dissemination of new skills throughout the organization. (The sociological and organizational impacts tend to be less expected than the technological impact, but can be even more significant.)

The technological impact includes the inevitable changes to the nature of work as well as the simple acceleration of existing work that will result from the spread of information technology that both follows and drives the achievement of the seamless environment. It is thus a three-step process: The first step, as implied above, is a mere speeding-up of work: The same work is done in the same way, but faster. The second step provides the first real change in corporate life: While it is still the same work, it is now done in dramatically new ways. Desktop publishing is a good example: We are still preparing the same sorts of documents as before, but most of the work is now done in the originator's office instead of in a central publishing office. The third step is the introduction of wholly new work: As work is done differently, new kinds of work will come to be done. A possible example of this in the publishing area is the introduction of *hypertextual* material, in which a document no longer consists of a serially accessible set of sentences, but a multiply-linked mass of material in several media.

As the elimination of seams makes computer-assisted work more amenable, it will spread more widely within the corporation, until it involves every professional, administrative, and clerical staff member. The potential for loss due to reduced productivity, redundant effort, misplaced files, inefficient or erroneous codes and spreadsheets, and improperly protected data, to name only a few areas, is enormous. (It is illuminating to estimate how much time is spent, institution-wide, just waiting for desktop systems to respond, even at rather high levels of performance; delays of even small fractions of a second, integrated over thousands of transactions on hundreds of workstations over a full day easily aggregate to appreciable staff-equivalents.) The point is that even though it is relatively straightforward, the technological impact is impressive.

Organizational impacts come somewhat more slowly, but as certainly. Paralleling the way in which the introduction of end-user computing is distributing the programmer/analyst function, the newer computer-assisted applications distribute other organizational functions. Many publishing and accounting functions, for example, are already moving from their central havens onto desktops all over the corporation. The central publishing department at my home institution now prepares less than one-quarter of the published papers (they used to prepare them all); for the balance, the entire process, including the preparation of camera-ready copy, is handled on desktop systems by the authors themselves or (*less* often) their secretarial staffs. It is no longer necessary to appeal to the Purchasing Department for a lease/purchase analysis, for anyone can do it in a few seconds with a standard spreadsheet macro. As these and other functions become widely distributed, the organization changes to track reality.

The sociological impact is concerned with the redistribution of power in the wake of the distribution of information technology. Much of the real power in any organization is wielded by those who control access to people and information. As seamless computing spreads through an organization, new ways of access will develop. In many cases, some or all of these ways will bypass the current guardians; as they lose control, some of their power and mystique will pass to those able to adapt to the new environment. As power shifts, relationships will change, often in unpredictable ways. (This general topic is far beyond the scope of this paper. It is given the attention it deserves in *In the Age of the Smart Machine: The future of work and power* by Shoshana Zuboff (Basic Books; 1988).)

Afterword: There are always seams

Before embarking this journey we should have admitted of its essential impossibility: There is no (permanently) seamless environment. In the first place, no two observers will have the same notion of what constitutes a seam. In the second, what constitutes a seam to a novice or infrequent user may cease to be a seam to that same person as an experienced user. The reverse is also true: Procedures that are thought by an experienced user to be rife with seams may not be considered seamy by a novice. Finally, and most importantly, I suspect, is the change in our perception of the possible that accompanies the elimination of each seam. Once we believe in the possibility of a desirable outcome, our impatience with the intervening hurdles increases: Our perception of the possible thus affects our perception of the necessary. As each new seamless tool is introduced into the workplace, we are tempted to add another style of work to our repertoire, or even another kind of work entirely. As we push the limits of the current environment, we encounter new seams. But the fewer there are, and the further out they are, the more productive will be our use of information technology.