Supporting End Users

The Organizational Interface: A Method for Supporting End Users of Packaged Software

By: Eileen M. Trauth
College of Business Administration
Northeastern University
Boston, Massachusetts 02115 U.S.A.

Elliot Cole
Center for School Study Councils
Graduate School of Education
University of Pennsylvania
Philadelphia, Pennsylvania 19104-7371 U.S.A.

Abstract
The growth of end-user computing and the trend toward the use of packaged software have focused greater attention on issues related to user support. A varied user population is requiring more tailored support, yet vendor-supplied software with a generic interface is increasingly being used. Four case studies illustrate ways in which organizational forms of support can enhance or overcome the limitations of the software interface. A framework called the organizational interface incorporates these organizational mechanisms by integrating the computer-human interaction (CHI), management information systems (MIS), and end-user computing (EUC) approaches to user support. This framework can be used in the design of end-user systems developed with packaged software and for retrofitting outdated user interfaces for an end-user computing environment. The organizational interface provides IS and end-user managers with a means to consider alternative methods of providing support for systems based on purchased software with fixed user interfaces.

Keywords: End-user computing, computer-human interaction, human factors, information centers, software interface, ergonomics, user interface, user support


Introduction
There are two current trends that have thrust user support to the forefront of IS management concerns. These trends demand that serious attention be given to determining which types of support are appropriate in certain contexts. The first trend is the dynamic growth of end-user computing. The increased number of users possess a wide range of characteristics—from the sophisticated to the novice, the motivated to the resistant, and the frequent to the intermittent. As a result, existing applications may have software interfaces that are no longer adequate. The reason may be an interface that was never really sufficient (but was overcome by the skill level of the users), or one that is no longer appropriate given the type and diversity of current users. Responding to the needs of this new class of users requires support that is more closely tailored to the individual in the context of use.

The second trend—the increased use of packaged software—complicates the process of providing appropriate support because system developers have less control over the development process. Whether systems are being developed by IS professionals or end users, they are increasingly being configured from purchased software. In such cases, software designed for the generic user is the basis for the application. Further, this software has a user interface that reflects certain assumptions about the user, such as knowledge level, degree of motivation, and problem-solving style.

Using hardware and software features, the vendor develops a user interface for generic users in generic contexts. The challenge to IS and end-user managers is to provide support for specific users in specific contexts. The purpose of this article is to help managers provide adequate user support for systems built around purchased software with fixed user interfaces. The article demonstrates ways in which organizational forms of support can be used as a substitute for
or to enhance software features in order to support the needs of an organization's users. These approaches are then configured into a conceptual framework of user support called the organizational interface.¹

Illustrations of Organizational Support

The following cases (called examples here) illustrate ways in which organizations have overcome the limitations inherent in the existing or packaged software through the use of organizational forms of support.² These four case studies show the structures, processes, and behaviors that were put in place to make technology more accessible to individuals. They demonstrate approaches that organizations can take to choose the appropriate user support mechanisms.

Example #1

Context

A multinational petrochemical company, Z (not its real name), was confronted with an issue common in many companies—information processing demand was exceeding supply. The natural response to this situation is to have the IS department offload work to the users. At Z, fiscal constraints prohibited a complete overhaul of the existing third generation computing environment. In such an environment, Z's end users would be confronted with a difficult user interface—one intended for use by computer professionals. Management feared that such a situation would diminish end-user motivation to use the system, increase errors during processing, and ultimately cause distribution of inaccurate information throughout the company.

Support Mechanisms

The solution to the dilemma was to make selected software changes supplemented by other types of support. An extract database and fourth generation language (4GL) tools were made available to facilitate access to corporate data. Users then had direct access to the corporate data needed in their work.

An extract database accesses virtually all transaction processing systems, as well as summary data on those transactions. At the same time, it protects the corporate database from problems that can arise from use by naive users. The 4GL tools constitute a new means of support that overcomes the barriers to data access that existed in the third generation environment. The use of a 4GL allows for data extraction, manipulation, and analysis; graphics; and system (re)design. Some of these tools can be used with only a few hours of training; others require more substantial learning time. The tools also contain features that correct errors in the command language and report specification.

At Z, the extract database and the 4GL tools facilitated direct user access to the data. However, the IS steering committee also recognized that these mechanisms alone would not be sufficient for all users because they ranged in skill level from novices to professional programmers. Therefore, a second decision was made to place the extract database and the 4GL tools under the domain of the information center, whose function is to conduct training sessions and staff a hot line.

Management at Z acknowledged that managers are discretionary end users and may lack the time or inclination to use the extract database and learn about 4GL tools. They determined that a third approach was needed for these individuals. A group of programmers called a Quick Response Team was established and given responsibility for developing time-sensitive ad hoc reports needed by line and staff management. This reporting method replaced the previous method, which involved a formal written request, management authorization, priority assignment, and scheduling—a process that would often take weeks. By using the Quick Response Team, a manager is now able to receive information from the corporate database within a few days simply by placing a telephone call.

Lessons

This case illustrates the use of organizational forms of support to enhance an unsatisfactory software interface. Since fiscal constraints pre-
cluded the replacement of the third generation interface, data (extract database), software (4GL tools), personnel (information center and hot line staff), and procedures (Quick Response Team and training) were used to overcome the interface's limitations. This example shows how a firm can provide alternate routes to corporate information. For certain individuals, the software interface enhanced by 4GL tools provides a second door to the company's computing resources. The availability of the extract database gives users greater access while protecting valuable data from inadvertent errors. The use of personnel such as the information center staff and the Quick Response Team takes into account the information processing needs and time demands of the range of users in the company.

By establishing an information center, Z provided a centralized location for user support through training, the availability of 4GL tools and extract database, and the hot line. Having these support features under its domain now allows the information center staff to monitor and respond to changing user needs.

**Example #2**

**Context**

In January 1987 Digital Equipment Corporation announced the DECwindows program, whose goal was to provide a modern, consistent user interface for workstation software. Because it was Digital's largest software development project to date, considerable effort by the Software Usability Engineering group was needed to coordinate the hundreds of people around the world who worked on the product. To promote consistency in the development of this product, two tools were created. One was the XUI (X User Interface) Style Guide, which describes the elements of the common user interface style to be used. The second was XUI Toolkit, a software development tool that implements the Style Guide recommendations by default. The default values of the object-oriented XUI Toolkit enable the application developer to produce interface components that conform to the Style Guide specifications.

During this project, Digital encountered several user support issues. The first issue was the importance of making the XUI Toolkit available to application developers as quickly as possible. However, waiting for a final version of the planned XUI Style Guide before beginning the development effort would delay work on DECwindows. An alternative was to reduce the amount of documentation included in the Guide. However, while this might shorten DECwindows' development cycle, such a reduction could possibly increase the time required for learning to use the XUI Toolkit and for debugging code developed with it, thereby also delaying the introduction of DECwindows.

The second issue was that no general purpose interface style (such as the one promoted by the XUI Style Guide) can anticipate all the needs of the many different applications being developed with it. A final issue was the difficulty of providing internal support for the application development team because team members were distributed throughout different locations in a variety of countries and across different time zones.

**Support Mechanisms**

Given the constraints mentioned above, the tradeoff was between holding up development of DECwindows until a complete style guide was available and allowing developers to begin using the toolkit immediately. Digital decided to make the XUI Toolkit available to developers before the final version of the XUI Style Guide was completed, but to provide other means of support as well. The approach was to supplement the "working version" of the Style Guide with support provided through use of electronic conferences available to Digital employees and implemented with VAX Notes software. These electronic conferences enabled users of the XUI Toolkit to submit questions and problems and allowed the Software Usability Engineering group to provide answers and solutions in a timely and useful fashion.

By the end of 1988, 12 separate conferences associated with the DECwindows development effort had been established. Usage data reported by Good (1989a, p.80) indicates that 2,089 participants engaged in discussions about 3,461 topics. The total number of notes was 17,901, with an average of 297 notes per week. (These

---

3 For a more detailed description of the development of DECwindows, see Good (1989a).
Supporting End Users

data describe usage in terms of the number of messages written. Since reading activity was not recorded, the actual usage was higher.)

In the course of using these conferences, there was also an unanticipated result: peers began to provide direct support to each other. It was intended that users experiencing a problem with the Toolkit or Style Guide would first search the conference notes to see if that issue had already been raised. If it had, the system would provide both an answer to the question and the name of another user who had faced the same difficulty. In this way, one user could discuss both the problem and the proposed solution with someone who had practical experience with the solution, and possibly knowledge of unforeseen and unanticipated effects. If this issue had not been raised previously, someone from the Software Usability Engineering group would provide the answer.

However, what actually occurred was that users began to provide unsolicited responses to queries entered into the conferences. Many of the users had developed considerable expertise with the XUI Toolkit. They were able to note relatively unusual problems that had arisen as a result of reaching the limitations in the tool’s functionality. To deal with these situations, users had discovered and developed patches and workarounds. As a result, support of the Toolkit moved beyond its developers and into the user community itself.

Lessons

This case, which centers around the substitution of personnel (peers and the Software Usability Engineering group) for formal data (completed system documentation), provides three important lessons about user support. The first is that organizational contingencies may require that one means of providing user support be substituted for another. That is, what would normally be conveyed through formal system documentation was instead (due to time constraints) communicated through electronic conferences. This also illustrates a way in which user support can be tailored to the specific needs of a specific class of users—in this case, the developers using XUI Toolkit.

The second lesson is that peers can be a valuable mechanism for providing user support. One who has encountered a similar problem in a comparable context may well be the best person to offer assistance.

The final lesson is that user support often comes about in unplanned ways. In this case, it was peers using the electronic network to answer each other’s questions. This suggests the need for management mechanisms to allow unplanned forms of support to develop and flourish.

Example #3

Context

Researchers at the University of Pennsylvania’s Graduate School of Education were asked to develop a tool to assist public school administrators in developing work performance measures and in conducting analyses of job responsibilities (Kowalski, et al., 1990). Once the tool was developed, an eight-session workshop was established to train users and to enhance their ability to develop new performance measures. To accomplish these objectives, workshop participants needed to learn how to retrieve data from flat files using dBASE III Plus. This software was chosen, in part, because it supports both a menu-driven interface for novices and a command-driven interface for more experienced users.

The goal of the workshop was to decrease the amount of time devoted to computer-related portions of the task in order to allow more time and energy for data analysis. To that end, substantial transparency was incorporated into the system startup process. For example, each computer automatically “booted” from power-up into dBASE III and opened a specific database file and index. The menu-driven interface was chosen over the command-driven mode because it is generally considered more appropriate for novices and occasional users of a software package. In order to accomplish their objectives, these school administrators needed to learn only six basic dBASE features that were distributed across four pull-down menus.

The objective of the first session was to convey a conceptual understanding of pull-down/pop-up windows that make up the dBASE menu. However, there were two problems in using the interface. The first problem concerned confusion about menu navigation and sequencing of operations. The second and biggest problem related
Supporting End Users

to the details of constructing “displays” of information. One slip of the cursor control key could eliminate several minutes of work without an opportunity for error recovery. In addition, the dBASE menus did not allow modification of a previous dialog sequence. Therefore, by the end of the second session, workshop participants were becoming frustrated with the software interface. It was clear to the workshop designers that some changes were required.

Support Mechanisms

In analyzing the situation, project staff noted that whereas only six commands were needed to accomplish the users’ objectives, the menu-driven interface presented them with about 50 commands, plus other options present in additional pop-up menus. That is, only about 12 percent of the menu’s commands were relevant to these users. They also noted that some of the participants had developed their own “cheat sheets” aimed at making the menus easier to use. Consequently, the project staff decided to abandon the novice-oriented menus and use selected features of the command-driven interface supplemented by context-specific documentation. Project staff believed that using the command-driven software interface in conjunction with specifically prepared documentation would result in five important benefits: (1) the application would be easier to learn; (2) the interface would be less confusing to the user; (3) better error recovery would be available; (4) the time required to enter a command would decrease; and (5) previous command strings could be edited, something that was particularly important for performing the workshop’s primary task.

The documentation was organized according to the sequence of steps involved in the analysis carried out by the users in accomplishing their task. Additionally, it incorporated examples of each command’s use and explanations of each command’s outcome. Finally, it made use of different typographical fonts to identify the different components of a command line.

Workshop participants found the resulting command-driven interface with customized documentation far superior to the menu-driven interface. With the command interface, participants developed an understanding of how the system worked, thereby decreasing learning time and increasing the amount of knowledge retained. The ability to easily revise the previous command line had several results. First, it decreased the amount of time to retrieve information. Second, it fostered data exploration because modifications could be made and executed in a matter of seconds rather than the minutes required for the menus. Third, it simplified the user’s view of the system.

Lessons

This case provides an interesting lesson about user support. What would normally be thought of as the more difficult interface (command-driven), ends up being the more appropriate “user friendly” interface (menu-driven) when combined with some organizational support (context-specific documentation). The designers of the user interface—the vendors—developed two ways of accessing a wide range of dBASE commands because they were designing for the generic user. The underlying assumption was that even the naïve user would want access to the full range of commands. However, in this particular context, the availability of so many menu commands hindered user accessibility. This observation is not a criticism of the software or the vendor. Rather, it is evidence that generic interfaces, no matter how “user friendly” cannot take all the individual and organizational contingencies into account. Therefore, the best support vehicle was one that at the outset might have seemed counterintuitive. What made the command-driven interface appropriate was that it allowed the user to focus on a subset of the commands. What made it practical was its use in conjunction with context-specific documentation.

Example #4

Context

Dublin City University, Ireland’s newest university, has experienced dramatic growth in end-user computing in the past few years. Today, all students in the university are expected to use the computer in their studies. A significant number of them use word processing for producing assignments and theses.
A problem recently addressed by the university concerned changing the campus-wide standard for word processing software. Students, faculty, and staff complained that the software currently in use, LEX, did not have an adequate user interface compared to what is currently available in other word processing software. At the time LEX was acquired, it was adequate for several reasons: (1) it represented the current state-of-the-art in user interfaces; (2) it was acquired when the university had only a mainframe computer; (3) it was affordable; and (4) word processing was not very pervasive on the campus. When the university began to introduce personal computers a few years ago, the decision was made to acquire the personal computer version of LEX. It was compatible with the mainframe, the price was right, and the few people doing word processing were familiar with that software.

The problem is that today, virtually everyone on campus does word processing. The addition of a new type of end user to the computing environment—one with little technological literacy—resulted in dissatisfaction with LEX. This was manifested in several ways. First, difficulty in using the software resulted in decreased productivity. Second, some students, faculty, and staff began to use other word processing software. As a result, multiple versions of software were circulating around campus, students were vulnerable to computer viruses acquired from copying software, and the computer center would not support these "unofficial" software packages.

Support Mechanisms

Upon investigation of this issue, consideration was given to changing the campus standard to WordPerfect. But there was a technical problem. State-of-the-art word processing languages such as WordPerfect require computers with a hard disk, yet most of the personal computers on campus were dual floppy machines. The cost of purchasing a new word processing package plus the additional cost of simultaneously upgrading all personal computers on campus with hard disks was beyond budgetary constraints.

The decision was reached to use a combination of software and personnel mechanisms that would gradually improve user support for word processing. The university agreed to change its word processing standard to WordPerfect in order to provide software with an improved user interface. But it decided to use a phased migration strategy. First, personal computers in student labs were networked so they could share a common hard disk, which could now hold WordPerfect. Second, computers in faculty offices were scheduled for gradual upgrading to install hard disks. Finally, the university established a policy whereby all personal computers purchased in the future would have hard disks. In the meantime, faculty and staff continued to use LEX. But during this transition period, increased personnel support was made available. At first, someone was available only a few hours a week to answer user queries. But now someone is available at all times during the work week to answer questions.

Lessons

This case illustrates a trade-off between software and personnel as user support features in the presence of fiscal and technological constraints. When organizational contingencies prohibited the university from immediately acquiring the most state-of-the-art interface, the school chose to substitute people for software while it gradually upgraded its facilities. The limitations in the existing software were overcome by the use of organizational features such as help desks and hot lines.

These cases represent four different ways in which organizational mechanisms were put in place to overcome the limitations of the user interface and thereby expand the notion of user support. In the first case, a petrochemical company overcame the limitations of an outmoded third generation interface by providing alternate routes to corporate information. These included some fourth generation tools, an extract database, support personnel, and several new organizational procedures. In the second case, a vendor responded to the time constraints associated with the use of a new development tool by supplementing incomplete system documentation with peer support through electronic networks. Contrary to conventional wisdom regarding support for novice users, the third case showed how some users were best supported by a novice-oriented, command-driven interface coupled with context-specific documentation. This documentation enabled the users to focus on the subset of commands relevant to their context. In so do-
Supporting End Users

ing, it eliminated the problem of information overload associated with the use of the menu-driven interface. In the fourth case, a university found that a phased migration to a new computing environment was the most realistic choice given technological and fiscal constraints. During the process of upgrading to software that had a better user interface, the university used personnel support as an interim solution.

The Organizational Interface Framework

All of these cases have one thing in common—they use organizational forms of support to enhance the existing user interface. This suggests that the concept of the user interface, a concept that embodies only hardware and software features, is but a part of the solution to the problem of providing adequate end-user support in today's computing environment. As an alternative, we present the organizational interface—a conceptual framework that brings together the range of support mechanisms described in the cases above. This framework represents the integration of three existing traditions of user support—computer-human interaction (CHI), management information systems (MIS), and end-user computing (EUC).

The CHI approach to user support

Along with the movement away from offline batch processing and toward interactive computing arose the need to study the human-machine interaction issues associated with online computer use. The field of computer-human interaction (CHI) has emerged in response to this need. CHI has its historical roots in a number of disciplines including human factors, physiological psychology, cognitive psychology, and computer science (Gaines and Shaw, 1986; Gerlach and Kuo, 1991; Newell and Card, 1985; Sondheimer and Ralles, 1982).

Support Mechanisms

Broadly defined, the CHI tradition is concerned with providing ease of use through software and hardware. The CHI approach to support is present in vendor organizations that build features into the software or user interface to assist the generic user. Such features include screen design, feedback and help, error control, and user dialogue. In some cases, vendors are also involved in the development of hardware features such as workstation design. Work on the design of the software interface has been aimed at increasing the efficiency of system use by focusing on speed and accuracy (Schneiderman, 1980). Work on the hardware aspects of computer-human interaction has been aimed largely at reducing physical stress symptoms such as eye, neck, and back strain (Sauter, et al., 1983). The objective of user support, reflected in the term user interface, is to join the human and technological systems together by placing hardware and software features between the user and the computer application (see Figure 1).

Definition of User

In order to understand the definition of the user reflected in the CHI approach it is helpful to know the history and orientation of the CHI tradition. It has its origins in human factors and began from the study of human-machine interaction issues related to aircraft pilots (DeGreene, 1970). These origins remain a strong influence on CHI practices today. The underlying assumption is a well-understood user engaged in a narrow range of tasks. The unit of analysis is a single individual in an asocial context. The intent is to facilitate the person's interaction with the technology, not to change the organization.

Assessment

A significant contribution of the CHI tradition is its goal of developing empirically supported guidelines for the design and evaluation of interfaces (Fried, 1982; Knittle, et al., 1986; Norman, 1983; Schneiderman, 1980). Three common elements can be found in these guidelines. First, the interface should be forgiving. That is, it should allow the user to recover from mistakes. Second, the interface should be transparent. That is, the language and logic of the computer-human interaction should not be so foreign to the user that it presents a barrier to effective communication. Finally, the burden of accommodation should be on the system and not the user. If successfully applied, these interface characteristics are ex-
pected to reduce the amount of formal training necessary to successfully interact with a computer-based system. Conversely, an expert user would receive the support he or she expects without being encumbered by features designed for the novice.

The three primary contributions of the CHI approach to user support are: (1) introducing an interface, a mechanism at the point of intersection between the human and technological (i.e. hardware and software) components, during system design; (2) responding to different cognitive styles through the provision of alternative software interfaces; and (3) identifying measurable performance criteria.

However, there are also weaknesses in this approach to user support. The first weakness results from the use of a generic user model. That is, while a software product may have alternative interfaces (e.g., both a menu and a command language), it is not intended to adjust itself to the particular needs of a specific user. Second, support is only provided through hardware and software features; organizational forms of support are not considered a part of the user interface. Third, the organizational context of use is not taken into account in interface design. As a result, the particular requirements of a specific organization are not addressed. However, organizational realities, such as those described in the cases, suggest that user support consisting solely of hardware and software interface features will be insufficient to provide the level of assistance needed. Financial constraints may prohibit an organization from obtaining state-of-the-art software interfaces. In addition, these interfaces may assume user characteristics that are not present in the organization. For example, the interface may assume an organizational setting with individuals who will use the software extensively and are therefore willing to tolerate a long learning curve. The user of an engineering workstation is such an example. The more likely case in business, however, is an intermittent and discretionary user who is under time pressure to begin using an application, a user who must use a variety of software tools, or a worker who is using insufficiently documented software.

For the reasons just given, user support provided by the CHI model has limitations that must be met by other forms of user support. These limitations are natural. Since the interfaces are created by vendors who have no control over the organi-

![Figure 1. The User Interface Joins the Human and Technological Systems](image-url)
zational support available, they naturally place it outside the scope of their user model. Table 1 summarizes this approach.

The MIS approach to user support
Both the business use of computer-based information systems and the value expected to derive from their use have grown significantly in recent years. The change from data processing to management information systems reflects the orientation away from a back room "shop," which carries out transaction processing activities, to a more sophisticated function that is able to provide assistance in managerial decision making and strategic planning. Thus, the objective of the MIS tradition of user support is effective use of information systems by personnel in the firm.

Support Mechanisms
From its beginning, the MIS literature has stressed the importance of addressing both the technical and organizational aspects of an information system. For example, Mason and Mitroff (1973) articulated the goal of considering not only technical issues but also psychological type, problem type, organizational context, and mode of presentation in the design of information systems. However, achieving this goal in the 1970s involved roles and players different from those that exist today. In the 1970s the software was generally designed by an in-house staff, systems were developed by IS professionals, and user involvement was limited to the use of pre-specified output. In today's environment, end users are increasingly developing their own systems. Furthermore, user involvement has grown from the receipt of periodic reports to hands-on interaction with systems. As a result, the goal of addressing both the technical and organizational aspects of information systems is being pursued, in large part, through the provision of user support.

The MIS approach to user support involves the use of organizational mechanisms to assist in the introduction of information systems into the organization. Providing support in the forms of training, documentation, and consulting staff facilitates the user's access to computer-based applications.

Table 1. CHI Approach to User Support

<table>
<thead>
<tr>
<th>Goal of Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facilitate individual’s interaction with technology</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Definition of User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single individual in associal context</td>
</tr>
<tr>
<td>Person engaged in narrow range of tasks</td>
</tr>
<tr>
<td>Individual characteristics well defined</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Support Mechanisms: User Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software features</td>
</tr>
<tr>
<td>Hardware features</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strengths</th>
</tr>
</thead>
<tbody>
<tr>
<td>User interface concept</td>
</tr>
<tr>
<td>Alternative software interfaces</td>
</tr>
<tr>
<td>Measurable performance criteria</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic user model</td>
</tr>
<tr>
<td>Support only through hardware/software</td>
</tr>
<tr>
<td>Context outside scope of interface design</td>
</tr>
</tbody>
</table>

Definition of User
The trend toward organization-wide use of information systems has resulted in a change in orientation toward system implementation and management. This change has been driven, in part, by the need for greater performance by the IS function because of an organization's increased dependence upon information technology (Bostrom and Heinen, 1977a; 1977b). The result has been greater emphasis on providing user support during system implementation and management (Leitheiser and Wetherbe, 1986; Nelson and Cheney, 1987). In this way, users are incorporated into the information processing infrastructure (Srinivasan and Davis, 1987). In the MIS tradition, the “user” can be an individual, a work group, or the entire organization (Leifer, 1988). In contrast to the CHI-world view, the user is not restricted to the terminal operator.

Assessment
The MIS approach to user support responds to two important limitations present in the CHI
model. First, it recognizes the importance of organizational contingencies and provides organizational mechanisms for responding to them. Second, by considering work groups as well as the individual, it places the technology in a social context.

However, this user support tradition has limitations as well, and they are a function of the way in which support is managed. First, in providing organizational forms of support during system implementation, they are viewed as part of the process of introducing the system rather than part of the system itself (Martin and McClure, 1988). They are, therefore, viewed as managerial actions taken at the end of the process, after the rest of the design is "fixed." When this happens, these forms of support cannot easily be considered as alternatives to the software interface features that are considered during system design and development. For example, whether a company has database intermediaries to help personnel access corporate data would not likely be a factor in designing a software interface to a database, or the criteria used to evaluate alternative database packages (Culnan, 1983).

Second, because this support tradition is closely linked to the implementation process, some forms of user support reside outside its administrative scope. Three examples illustrate this point. While development, operations, and management of corporate databases may be part of the corporate IS function, database intermediaries may work in the corporate library or information center and report to another administrative area. When human factors or CHI personnel exist in a firm, they generally work with the system design team and not with those concerned with implementing organizational forms of support.

Third, personnel staffing user support hot lines are normally part of the operations group, and communication between them and development personnel may be organizationally difficult to achieve. Thus, while the MIS approach broadens the scope of user support, it does not facilitate trade-offs between software and organizational support features. The MIS approach is summarized in Table 2.

The EUC approach to user support

The emergence of the term "end user" marks the spread of computing outward into the organization. The term "end-user computing" (EUC) suggests that computer-based applications are available not only for management needs but for addressing the complete range of information processing tasks of an organization (Beath and Ives, 1986; Hirschheim, 1985; Keen and Scott Morton, 1978). Another term more recently used to describe this phenomenon is "desktop computing." The 1970s and 1980s witnessed considerable growth in end-user computing as compared to traditional computing (Benjamin, 1982; Rockart and Flannery, 1983). This growth was due to several factors including: the desire for greater end-user control and flexibility in using applications; the technical/economic feasibility of information technology use by non-IS personnel; business conditions requiring increased information within a compressed timeframe; lower development costs associated with end-user developed applications compared to lengthy applications development backlogs within IS departments; and users' increased awareness of the potential and capabilities of computers because of widespread computer literacy (Alavi, 1985; Leitheiser and Wetherbe, 1986; Rockart and Flannery, 1983).

Several researchers have documented the growth stages of end-user computing, drawing parallels to Nolan's stage theory (for example, Henderson and Treacy, 1986; Huff, et al., 1988). They observe that as end users move from simple, stand-alone applications to more sophisticated, integrated ones, management structures must change as well. A consistent theme is the need to tailor the support infrastructure to the evolving needs of users.

Support Mechanisms

From its inception, support has been an integral part of end-user computing. Recent studies of EUC management have found that user support is directly linked to user satisfaction and represents a major concern for EUC managers (Guimaraes and Vasudevan, 1986; Rivard and

---

4 Kling and Iacono (1986) consider the term desktop computing to be more specific than the term end-user computing. They view the former as a more accurate description of the computing power available to people in the workplace. In this paper, the term end-user computing is meant to include desktop computing as well.
Supporting End Users

Table 2. MIS Approach to User Support

<table>
<thead>
<tr>
<th>Goal of Support</th>
<th>Definition of User</th>
</tr>
</thead>
</table>
| • Effective use of information systems within an organization | • Single individual/work group/entire organization  
• Part of information processing infrastructure |

Support Mechanisms: Post-Implementation Support

| • Training  
• Documentation  
• Internal consultants |

Strengths

• Takes organizational contingencies into account  
• Technology placed in social context

Weaknesses

• Support not viewed as part of system  
• Cannot make tradeoffs between software interface and organizational forms of support  
• Organizational forms of support administered by different parts of the company

Huff, 1988). However, the type and extent of support has paralleled the growth stages of EUC itself. In the early 1980s IS departments maintained a "hands off" policy in which support consisted solely of equipment demonstrations or assistance with installation; the user was left to learn by trial and error or by reading manuals. By the middle of the decade, support had grown to include help desks, training, personal computer libraries, and, to a lesser extent, user groups (Benson, 1983; Guimaraes and Vasudevan, 1986).

Support for end-user computing today includes a wide range of services such as: consulting, technical/product support, user hot lines, help desks, training, and user groups (Leitheiser and Wetherbe, 1986; Rivard and Huff, 1988). The most common formal mechanism for providing user support is through the information center (IC). Originally consisting of a few IS personnel charged with assisting mainframe users with fourth generation languages, the IC is now an organizational unit whose function is to facilitate and coordinate end-user computing by offering a range of support services.

In addition to formal mechanisms, there are interpersonal support means. During the early stages of end-user computing in an organization, support generally occurs informally through user/IS contacts or between novice and experienced users. As end-user computing matures in the organization, this support becomes more formal (Huff, et al., 1988).

Research has shown that often the first line of support is provided by one's peers (Cole, 1984; Lee, 1986). Even where formal support exists, end users rely heavily on informal mechanisms within work groups (George, et al., 1989). This support can take several forms. At its most informal level, interpersonal support is provided to colleagues by an individual in a work group who is experienced in the use of an application. Since this individual understands its use in a specific context, he or she can offer support on an individual basis.

A more formal mechanism for peer support is provided through electronic notes or conferences. For instance, people who encounter difficulties
Supporting End Users

with an application can ask questions via an electronic messaging system, and others who have encountered similar problems can offer assistance.

More formal still is the establishment of a liaison position in work groups. A designated individual functions as a bridge between the IS department and the end users. A channel of communication between the system developers and the system users is thereby established to respond to individual support issues.

The EUC approach to user support can be viewed as utilizing people to help the user with an application. The development of peer support and user networks reflects the inherently social nature of end-user computing and the accompanying need to account for it in the design and management of the end-user computing environment (Kling, 1987; Lepore, et al., 1989).

Definition of User

There is a certain amount of variation in defining the user according to this support tradition. At one extreme is a definition that includes programmers, users, and support personnel (Rockart and Flannery, 1983). At the other are users who are application developers (Rivard and Huff, 1988). In between, and the most common, are the non-IS professionals who use (and sometimes develop and manage) computer-based applications to support their work in some functional area of the firm (Alavi, 1985; Benson, 1983; Sipior and Sanders, 1989). Like the IS view, the user may be defined as either an individual or as a work group.

Assessment

The main strength of this support tradition is its recognition of the inherent variability in the user population. The literature consistently calls for a range of user support mechanisms tailored to the range of individual needs in an organization. Another strength is the dynamic nature of user support. That is, many organizations recognize that support needs change along with individual and organizational learning curves. Finally, when user support personnel are moved closer to the functional areas, user support reaches out to the user in his/her workplace (Henderson and Treacy, 1986; Huff, et al., 1988; Rockart and Flannery, 1983).

One potential weakness of this approach is related to peer support. Not all organizations have placed support personnel in functional areas. Consequently, users may need to rely on informal sources for help. Because it is unplanned and undesigned, this type of support depends upon the willingness of individuals to help each other or the value placed upon helping others in a particular work group. The quality of peer support is also highly variable. At best, it is relevant support in context; at worst it is "the blind leading the blind."

A second weakness is that not all the available support personnel in an organization are incorporated into the EUC view of support. For example, the database intermediary or "chauffeur" (Culnan, 1983) whose job is to access databases for others is not included under this view.

Finally, there is no evidence that these support services play an influential role in decisions regarding software acquisition and use. Thus, as with the MIS approach, support is considered at the end of the system development process rather than during application design. This approach is summarized in Table 3.

The organizational interface approach to user support

This approach uses the term "interface" because it borrows from the CHI tradition the notion of positioning user support features between the user and the application—at the point of interaction or interface between the human and the technological systems. It uses the term "organizational" because it adds the organizational mechanisms present in the IS and EUC approaches to the software features of the user interface. The framework is depicted in Figure 2.

Support Mechanisms

As Figure 3 shows, the organizational interface considers each of the components of the information system as an option to be positioned between the user and the application in the provision of user support. The hardware interface includes ergonomic features such as lighting, input devices, furniture, and workstation design. The software interface includes such features as screen design, menus, windows, online...
Table 3. EUC Approach to User Support

<table>
<thead>
<tr>
<th>Goal of Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Individual control of personal computing environment</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Definition of User</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Non-IS professional</td>
</tr>
<tr>
<td>• Uses (sometimes develops and manages) computer-based applications</td>
</tr>
<tr>
<td>• Works in functional area of the company</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Support Mechanisms: Information Center and User Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Consulting</td>
</tr>
<tr>
<td>• Hot line/help desk</td>
</tr>
<tr>
<td>• Training</td>
</tr>
<tr>
<td>• Technical/product support</td>
</tr>
<tr>
<td>• Peer support (user groups/networks)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strengths</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Recognizes variability of user population</td>
</tr>
<tr>
<td>• Support provided at workplace level</td>
</tr>
<tr>
<td>• Wide range of support options</td>
</tr>
<tr>
<td>• Dynamic nature of support given</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Informal nature of peer support</td>
</tr>
<tr>
<td>• Support features not influencing software acquisition decisions</td>
</tr>
</tbody>
</table>

documentation, and help. Examples of the data interface are system documentation, electronic conferences, and the creation of extract databases. The procedures interface refers to corporate rules, procedures, and policies put in place to facilitate user access to systems. One such procedure is the establishment of an information center that could provide users with an identifiable source for help and information. Another procedure is training. Both users and IS staff contribute to the personnel interface—users through formal and informal networks, and IS professionals, such as database intermediaries, internal consultants, and hot line advisors, by providing individual support about particular problems.

Definition of User

In this approach to user support, the user is depicted as an independent end user working in a functional area of the firm. This individual not only uses an information system but might design, develop, and manage it as well.

Assessment

As Figure 3 shows, the organizational interface expands the CHI approach, which can only position a user interface (hardware and software features) between the user and the application. Organizational support activities are added to the interface design and can then be considered during the design of information systems based around purchased software.

This approach expands the MIS and EUC approaches in three ways. First, it formalizes mechanisms currently in place but not usually considered until system development is completed. Second, it recognizes informal peer support as a key component of the overall support
Implications of Using the Organizational Interface Framework

This framework is intended to be used in two situations. One is the design of end-user systems developed with packaged software. Organizational forms of support such as context-specific documentation, peer support through online conferences, and consultation from the IS staff can be employed to enhance the generic software interface by providing support geared to the specific needs of a particular group of users in the firm.

The second situation involves retrofitting an outdated user interface for a new computing environment. Applications based on third generation software are generally not suited to the knowledge/experience level of today's end users because they were intended to be used by IS professionals. While the natural solution might ap-
Supporting End Users

We acknowledge that one or more of these support features may currently exist in an organization. Nevertheless, the contribution of this framework is its suggestion that these organizational support features be considered in tandem with the software interface during system design. If organizational forms of support can be considered as alternatives to software features, then system designers, whether IS professionals or end users, can entertain trade-offs in light of organizational realities and constraints. However,
there are some implications associated with the use of this framework.

**Implications for system design and development**

One implication of using this framework is the change in what constitutes system design. While support in the form of the software interface is taken into account during system design, other forms of user support are normally considered during system implementation and management, if they are considered at all. Therefore, the use of the organizational interface as a support framework means that system designers, whether they are IS professionals or end users, will need to expand their definition of system design to include consideration of organizational forms of support. This is necessary so that the user interface accompanying vendor-provided software can be evaluated in the context of other support that can be made available. This broader view of design, one which incorporates both technological and organizational features, is consistent with other views of system development that have been expanded to incorporate behavioral and organizational considerations (for example, Bostrom and Heinen, 1977a; 1977b; Hirschheim, 1985; King and Kraemer, 1985; Mantei and Teorey, 1989; Markus, 1984; Mumford and Weir, 1979; Robey, 1987).

A change in perspective suggests implications for the education and orientation of system designers/developers as well. Systems developers working in the IS department would need to consider support features such as training and help desks, items often thought to be someone else's domain, during software evaluation and acquisition. End users who are developing systems may need to think in more formal terms about support features that had previously been unplanned and informal.

**Implications for organizational structure**

The use of this framework also has implications for a firm's organizational structure. This framework argues that all support functions—user interface design, ergonomics, database intermediaries, documentation, internal consultants, information centers, electronic conferences, and training—should be seen as alternatives to each other. However, these functions are often carried out by several different organizational units. Therefore, in order to provide a forum for considering these alternatives during system design, a company's organizational structure might need to be modified, or mechanisms for communication and coordination might need to be established.

**Implications for vendor organizations**

A final implication of the organizational interface approach to user support is not for the organizations employing the software, but for the vendors providing it. Consistent with the CHI approach, the vendor's contribution to user support has historically occurred through the software interface. To this end, human factors personnel have focused on improving software usability. Criteria for evaluating usability have come primarily from laboratory research. Recent trends, however, suggest that vendor organizations are expanding their view of support beyond usability testing to include customer perception of relevance within the work context (Good, 1989b). Several examples of this trend were reported at Digital Equipment Corporation (Wixon, 1989). One is that the company has placed a greater emphasis on field and customer testing of new products. Another is the increased role of service in the company's revenue mix, as well as the development of software tools that are customizable within the client's work context. These observations have led Wixon to conclude that not only users, but also vendor organizations, need to broaden their conception of user support to include a better understanding of the context within which the products will be used.

**Conclusion**

The framework presented in this paper provides a means of integrating the full range of user support mechanisms available in an organization so that trade-offs can be considered for overcoming the limitations of the existing software interface. By drawing together the CHI, MIS, and EUC approaches to user support, the organizational interface approach presents managers with a set of alternatives from which to choose in tailoring support to specific users in specific contexts. The organizational interface can then help managers
respond to the user support demands of today's computing environment. While one or more of the support mechanisms described in this paper may currently exist in an organization, this framework offers a way of bringing them together so that they can be viewed during system (re)design as alternatives or enhancements to the user interface accompanying purchased software. Thus, the notion of user interface is broadened to include organizational support features as well as software components.

Acknowledgements
We would like to thank Dennis Wixon from Digital Equipment Corporation and the faculty and staff at Dublin City University for their assistance in our research.

References

About the Authors
Eileen M. Trauth is an associate professor of management information systems at the College of Business Administration, Northeastern University. The topics addressed in her research include the impact of end-user technology, information resource management, telecommunications policy, and the global information services industry.
Her articles have appeared in such journals as ACM Transactions on Office Information Systems, Journal of Systems Management, Information and Management, and Telecommunications Policy. She is also the author of Information Literacy: An Introduction to Information Systems. During 1989-90 she was a Fulbright Scholar at Dublin City University in Ireland. She received her Ph.D. from the University of Pittsburgh.

Elliot Cole is a research associate at the Center for School Study Councils in the Graduate School of Education, University of Pennsylvania. His research and consulting focus on the interaction between users and their computing environments, concentrating on user mechanisms for customizing the environment. He applies this perspective to problems in end-user computing, office automation, and software maintenance. He is also president of The Institute for Cognitive Prosthetics, which customizes computing environments for brain-injured individuals. Before joining the staff at the University of Pennsylvania, Dr. Cole taught at Drexel University. He received his Ph.D. at Syracuse University.