

Designing Operations Systems Interfaces That Are Easy to Use

Nuray A. Aykin
James P. Cunningham
Joseph P. Rotella

An important factor in the planning and development of AT&T operations systems is their ease of use. This paper describes the value of easy-to-use operations systems, and what needs to be done to produce such systems. Projects need to implement systematic user-centered usability engineering processes, including setting measurable usability objectives, documenting user tasks and scenarios, prototyping iteratively, conducting structured design reviews, testing for usability, and examining post-delivery usability data. For operations systems, there needs to be an explicit focus on user interface consistency, both within and across systems. Consistency can be achieved by identifying common elements, collaboratively designing those elements, producing common specifications, and supporting the entire process with common tools, reusable software libraries, and centralized coordination. In addition, several other interface design principles are described that, if followed, will improve system usability.

Introduction

An operations system is easy to use if people can quickly learn to use it, accomplish their tasks efficiently, make few errors, remember how to use it, and enjoy using it. A system that is hard to use leads to customer dissatisfaction because users are frustrated, time is wasted, errors can cause lost revenue, and maintenance costs are high. The ease of using an operations system is determined by all aspects of the system that affect the user, including the human-computer interface, documentation, training, work context, physical characteristics, and, of course, the functionality of the system. All of these aspects must be carefully planned and developed to produce an easy-to-use operations system.

Advances in technology make it possible to develop operations systems that are easier to use today than were the systems of 10 years ago. For example, modern graphical-user interfaces are more suited to operations systems than character-based interfaces.¹ The use of artificial intelligence techniques can automate some functions that otherwise would burden the user. Documentation and training can be integrated with the

system and delivered on-line. Functions that used to be performed by separate operations systems can be integrated into a single system. Technological advances can help make an operations system easy to use, but technology alone is not enough. It is easy to develop a system that employs these technological advances and is still very difficult to use. In this paper, we briefly describe the benefits of making systems easy to use, and then describe processes and principles that can help make it happen.

Why Invest in Ease-of-Use?

Producing easy-to-use operations systems benefit both the customers and the producers of the systems, and the modest investment required will be paid back many times over.

Benefits for the customers include:

- User satisfaction—Well-designed user interfaces increase user satisfaction, creating less frustration, less fatigue, lower worker turnover, and a greater enthusiasm on the part of the user for learning about the system.
- Improved productivity—Well-designed

user interfaces eliminate unnecessary data entry and reduce task completion times, resulting in higher output per work-hour and per employee.

- Reduced error rates—User errors caused by difficult-to-use interfaces can adversely affect the communications service provided to end-customers, resulting in lost revenue.
- Reduced training—Well-designed and consistent interfaces, with integrated documentation and training, make systems easier to learn. Users can work with more flexibility on a variety of systems, without extensive cross-training, if the systems have familiar, consistent interfaces.
- Reduced support costs—Users will be more self-sufficient, reducing the need for support staff to help users through problems or difficulties.

Organizations that produce operations systems also benefit from building in ease-of-use by increasing revenue and lowering costs:

- Increased sales—The most obvious benefit is satisfied customers, and the resulting increase in sales.
- Decreased training costs—Systems that are easy to learn require less time for training customers, and marketing and support personnel.
- Decreased service and maintenance costs—Fewer calls to support organizations, and fewer requests to fix apparent system problems that are really caused by poor interface design, will result in lower service and maintenance costs.
- Shorter development cycle—The use of common interface designs, common tools, and reusable objects can shorten the development cycle. Systematic user-centered, usability engineering processes can reduce the amount of rework caused by interface defects.

How Much Should Be Invested?

There is no simple formula that will result in operations systems that are easy to use. But, since modern software systems with graphical user interfaces usually have about 50 percent of their lines of code devoted to the human-computer interface, it is worth a significant investment to ensure that those lines of code are contributing to customer satisfaction. A recent industry survey² of 31 moderate- to large-scale software development projects showed that an average of 6 percent of the budget was spent on the sort of usability engineering processes described in this paper. So, to keep up with the

competition, projects should have 5 to 10 percent of their resources devoted to ensuring ease-of-use.

Planning for ease-of-use must start at the very beginning of a project and continue throughout the entire product realization process. Three basic recommendations for planning and developing easy-to-use operations systems are:

- Emphasize user-centered usability engineering processes,
- Focus on consistency, and
- Follow basic interface design principles.

The remainder of this paper describes these recommendations in more detail.

User-Centered Usability Engineering Processes

The most important thing that a project can do to produce an easy-to-use operations system is to implement thorough user-centered usability engineering processes. If the correct planning and development processes are employed, the user interface will naturally follow basic design principles, and will delight the users. The key is to have an early and continual focus on the users of the system. At the beginning of the product realization process questions, such as the following, should be answered:

- Who are the users?
- What is the work environment like?
- What are the users trying to accomplish?
- What other systems do the users work with?

Users should be consulted continuously throughout the design, prototyping, and development process. During this process, designers should be asking:

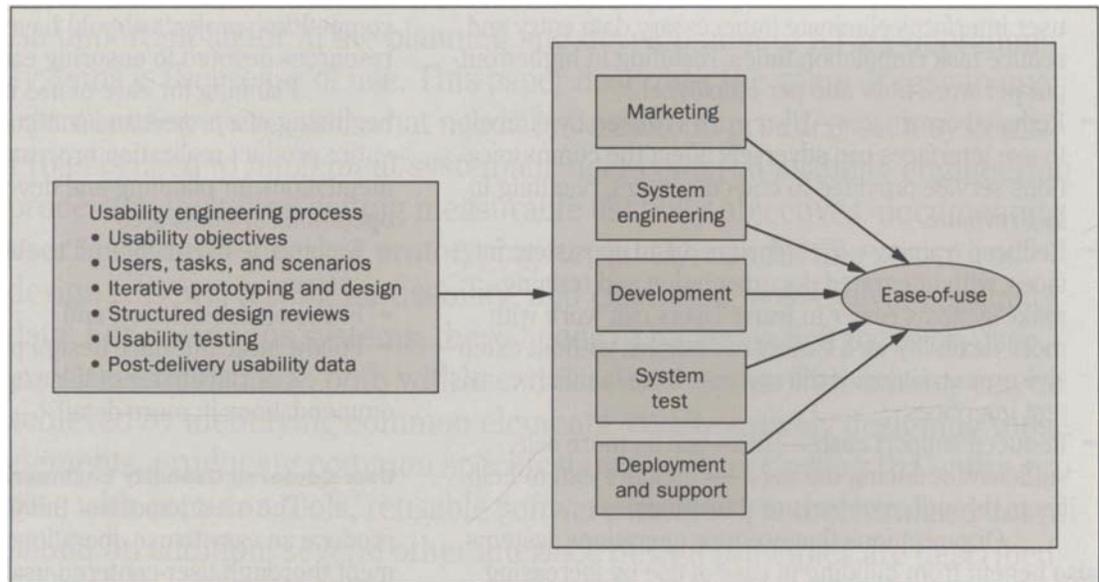
- What do users like and not like about the interface?
- Where do most errors occur?
- How long does it take to accomplish each task?

After the system is delivered, the users' views should still be considered:

- What do users complain about?
- What causes the calls to customer support?
- What suggestions for improvements are users making?

In short, to produce an easy-to-use system, the product realization process should be user-centered. The user's needs and system usability should drive all actions and decisions throughout the process, during marketing, systems engineering, development, testing, and deployment and support (see Figure 1). This section describes the most effective usability engineering processes.²⁻⁴

Figure 1. To produce an easy-to-use system, the product realization process should be user-centered. The user's needs and system usability should drive all actions and decisions throughout the process, during marketing, systems engineering, development, testing, deployment, and support.



Usability objectives. Set explicit usability objectives at the beginning of a project. The objectives should be stated in terms of something that can be measured—for example, a percentage of tested users that make no errors on a particular task, the average time to complete a particular task, or measurements of user satisfaction on a pre-defined rating scale. The objectives also should clearly state what level of the measured variable will be satisfactory. The nature of the objectives will be determined by the areas that are most important to the project—efficiency, error rate, ease of learning, etc. If the managers of a project put as much importance on meeting usability objectives as on meeting cost and schedule objectives, the result will be the employment of good usability engineering methods throughout the project.

Users, Tasks, and Scenarios. A description of the users and their characteristics should be documented and validated with the customer at the beginning of a project. Design the interface to match the skill level and experience of the users. A detailed list of the tasks and subtasks to be performed by the users should be compiled, and step-by-step scenarios of how the users will complete the tasks should be constructed. Validate the tasks and scenarios with the customer.

Iterative Prototyping and Redesign. Construct prototype user interface designs as early as possible in the product realization process, and then gradually refine the designs, based upon the feedback received. The first

prototypes can be drawings on paper. Then, rapid prototyping tools can be used to build systems that closely approximate the final product. Feedback on the prototypes should be obtained from all stakeholders—users, customers, engineers, product managers, etc. The feedback can be obtained by the two methods described next—*structured design reviews* and *usability testing*. At each stage, redesign the prototype based upon the feedback.

Structured Design Reviews. An efficient way to get feedback on a user interface design is to hold a meeting at which the design is reviewed in a structured way by a small group of stakeholders. (Another name for this type of review is “heuristic evaluation.”²) Design problems and suggested changes in the design should be identified and documented, but, for the sake of efficiency, comments should be accepted for later evaluation, and not be discussed or argued about at the review. The effects of the review on the design should be tracked. Design reviews can be held repeatedly during the design process.

Usability Testing. Have the users run through representative task scenarios using the interface. Measure the users’ performance, record their opinions, and compare the results to the usability objectives. Usability testing can take a variety of forms. For example, the testing could use many participants to collect detailed data for detailed analyses; or the testing could involve videotaping a few users and watching the tapes for representative problems. The type of testing will depend upon the time

and resources available, but it is important that some form of usability testing be done. The most useful design feedback comes from seeing users making errors or failing to complete tasks because of faults found in the user interface design.

Post-Delivery Usability Data. After an operations system is delivered to a customer, there are often opportunities to collect valuable usability data. Sometimes real users can be interviewed, given questionnaires, or observed while doing their jobs. Calls to a help desk or support line also can be analyzed to see if user interface defects are contributing to problems. In addition, AT&T personnel who have direct contact with customers can record customer comments or complaints. Information from all these data sources can be used to reevaluate the design for the next release or system.

Focus on Consistency

While user-centered usability engineering processes will lead to individual applications or systems that are easy to use, most users of operations systems employ multiple applications or systems to perform their jobs. For users who switch from system to system, having a consistent user interface across applications or systems is particularly important. Thus, extra effort should be devoted to achieving consistency.

Sometimes multiple applications are integrated, and can be accessed from a single workstation. Providing user interface consistency across these applications makes it easier to integrate the applications into what appears—to the user—to be a single, seamless system. The applications can act as modules that can be combined in a variety of configurations to meet a variety of customer needs.

Elements that Need to be Consistent. It is particularly important that elements that perform similar functions in different operations systems have consistent interfaces, since differences will adversely affect user performance. Elements that are often found in operations systems include:

- Basic tasks—Tasks such as printing, reading electronic mail, and saving records should be performed in a consistent way across systems.
- Maps—Status and configuration information displayed on network maps should be consistent in appearance and in the ways that users interact with them to obtain more information, for instance, or alter parameters.

- Forms—Information is often entered into forms that should be consistent in look, inter-field navigation actions, placement of common information, default handling, etc.
- Lists—Multi-column list displays, often used for alarms lists, activity logs, trouble-ticket logs, work assignments, etc., should have consistent headers, column dividers, scrolling techniques, etc.
- Graphical symbols—Network elements, office locations, types of nodes on network maps, etc., should be represented by common graphical symbols across operations systems.
- Terminology—Names, and especially acronyms, for common operations and objects across operations systems should be identified, defined, and employed consistently across all operations systems. Creating new terminology or using synonyms only confuses users.
- Fonts—Text fonts not only affect the readability of the text on a screen, they also play an important role in providing a consistent appearance across systems. Font specifications for specific activities, such as for labels, user-entered text and data, help text presented to the user, etc., should be consistent across systems.
- Colors—Colors used to code important information presented to the user, such as alarms, should be specified and consistent across operations systems. Other colors should be used consistently to improve the visual appeal of a system.
- Documentation—User documentation refers to both printed and online material, including manuals, online help, online documentation, and online messages. Documentation across systems should be consistent in appearance, format, and content, and should be consistent with the rest of the interface.

Methods for Producing Consistency. Achieving user interface consistency across applications requires hard work and attention to detail. When the applications are developed by different organizations, the task becomes even more difficult. However, experience with AT&T operations systems has shown that there are several methods that make it easier to achieve interface consistency:

- Common specifications—There should be a single interface specification for all applications. That specification should contain as much detail as possible, and should continue to evolve during the design and development process as more common design

details are agreed upon. The specification should refer to, and be consistent with, all relevant interface standards and guidelines, including international, national, corporate, and business unit-specific standards.

- Co-design—Interface features that are common across applications should be cooperatively designed by the interface designers of all the applications. Avoid, for instance, picking one application's design and calling that the standard. The results of the co-design process should be documented in the common specification.
- Frequent reviews—The interface designers for the various applications should review each others' designs to identify and resolve inconsistencies. These reviews should be frequent, so that potential inconsistencies can be caught before they become expensive to change. In addition, formal reviews of compliance with the common specification should be made periodically, with the results made available to the relevant managers.
- Common tools—All applications should use the same interface software development tools. This will enforce consistency in some aspects of the interface, and make it easier to develop consistency in the remaining aspects. The AT&T BaseWorX™ applications platform provides an architecture, a set of integrated software components, and customer-support services.⁵ The primary domain of BaseWorX is network management systems and operations systems that have common elements, such as user interfaces; networking; database management; interprocess communications; and application operations, administration, and maintenance. The BaseWorX platform provides reusable components and protocols to support these elements and to help application developers achieve consistency, while improving the quality and time-to-market of a product.
- Reuse—Entire software modules should be reused across applications whenever possible. This reuse can lead to the best consistency, and also result in the development of more reliable and lower-cost software. Objects that appear in many operations systems (maps, symbols, list displays, etc.) should be designed and developed once, and then put in a library to be shared across projects.
- Centralized coordination—The methods listed above can only work if the design and development of the user interface is centrally coordinated to facilitate communication among organizations and individuals,

maintain the common specification, conduct compliance reviews, and provide an information resource for the common tools and reusable modules. The coordinators work with the human factors engineers assigned to the individual systems, helping them produce consistent interfaces. The ease-of-use of each individual system is the responsibility of the human factors engineers for that system.

Other User Interface Design Principles

In addition to an emphasis on consistency, there are a number of other design principles that can help produce an easy-to-use system, if followed during the interface design phases of the usability engineering process. The particular set of principles given here is based on information in international standards,⁶ books on user interface design,^{7,8} and experience designing and evaluating interfaces for AT&T operations systems (see Figure 2). The descriptions of the principles given here are very brief, but the reader can refer to the references for more complete descriptions.

Adherence to many of these design principles can be evaluated during the structured design reviews that are part of the usability engineering process. Each principle can be assigned to a specific reviewer, who then has the responsibility of evaluating whether that principle is being followed.

Simplicity. Emphasis here should be on functional simplicity:

- Minimize the number of steps required to do a task.
- Minimize the number of windows accessed to complete a task.
- Minimize the number of menus in a hierarchy that must be displayed before an action can be invoked.
- Employ simple wording for messages and other information.
- Eliminate screen clutter.
- Provide the user with a pointing device to accomplish most functions. For example, transfer data by dragging a visual representation of the data to a visual representation of the destination—rather than by typing.
- Use the task list, scenarios, design reviews, and usability tests to identify the “bells and whistles” that can be eliminated.

Visual Appeal. People are more satisfied using human-computer interfaces that are visually appealing.

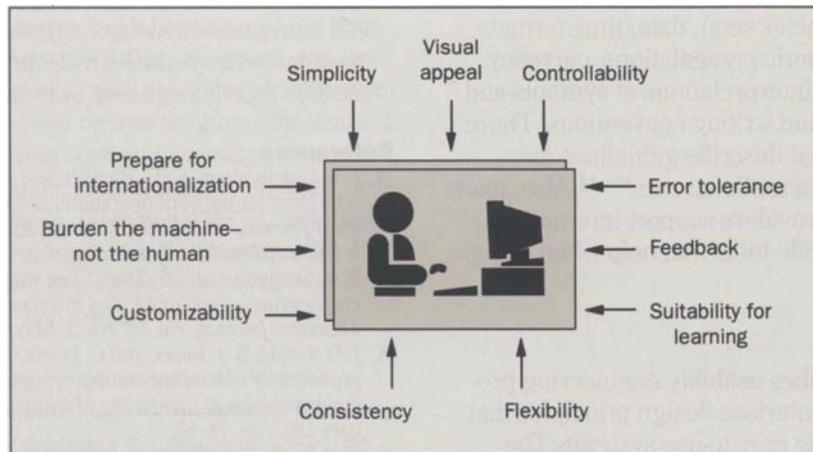


Figure 2. In addition to an emphasis on consistency, these design principles can, if followed during the interface design phases of the usability engineering process, help produce an easy-to-use system.

The clean, attractive looks of current popular graphical user interface environments, such as Apple's Macintosh,* Microsoft Windows,* Hewlett Packard's VUE,* and Sun's OpenWindows,* are largely due to the work of graphic designers, not human factors engineers or software developers. The services of a graphic designer can make an enormous difference in the look of an operations system.

Controllability. The user should be in control of the interaction, with the ability to start or interrupt the task at any time, access help, control the format of data display, and control the speed of interaction.

Error Tolerance. Make the system robust to human error. Small errors should not cause major unintended consequences, and users should be instructed how to correct errors.

Feedback. Keep the user aware of what is going on in the system, and provide timely feedback for all user actions. The system should inform users about the steps necessary to cause some desired outcome, or to prevent the unwanted consequences of an action.

Suitability for Learning. Design the interface to match the skill level and experience of the users. Guide the user through the learning stages, and allow for the natural progression from novice to expert. Provide prompts, dynamic tips (analogous to the "wizards" that are provided with Microsoft Excel*), and on-line help. Integrating documentation and training into the operations system can help bring new users up to speed quickly, and ensure that all the latest system features are covered.

Flexibility. Make the system flexible enough to fit the requirements of different tasks and the needs and skills of different users. Providing multiple ways of completing similar tasks permits users in different task environments to choose the ways that are most efficient for their environments.

Customizability. Build in the ability to customize the interface for the specific needs and tasks of a particular user. This permits a user to be more efficient, and allows an operations system to fit in gracefully with the work flow of any customer.

Burden the Machine—Not the Human. Automate as many functions as possible. The user should not have to perform any function that could be performed by the computer system. The automation may be very simple, such as automatically filling in a field with the most likely value, based on previous field entries. Or it may be more complex, such as the expert systems techniques used by TOPAS-ES to localize network faults,⁹ or the use of "intelligent agents" to automate information gathering tasks.

Internationalization. AT&T operations systems are being deployed in countries around the world. International customers expect operations systems that are customized for their own use—user interface and documentation in their language, date and time formatted according to their convention, etc. Early in the product realization process, the interface designers and software developers need to be aware of relevant standards, requirements, and cultural conventions for the countries where the systems will be deployed. Topics to consider include keyboard layouts, multinational language

specifications (e.g., character sets), date/time formats, collating sequences, numeric presentations, currency representations, cultural interpretations of symbols and icons, writing direction, and writing conventions. There are a number of books that describe guidelines for designing products for international use.¹⁰⁻¹² Also, many hardware and software providers support international software design and provide tools that help when designing for international use.

Conclusion

This paper describes usability engineering processes and general user interface design principles that can be helpful in achieving easy-to-use systems. The usability engineering processes should be user-centered, with user needs and usability considered at all stages of the product realization process. Particular emphasis needs to be placed on interface consistency, since there are numerous operations systems being developed within AT&T, and a comprehensive strategy is needed to achieve consistency across all the systems. Consistency can be achieved by identifying the common elements (network maps, forms, symbols representing network elements, etc.), collaboratively designing those elements, producing common specifications, and supporting the entire process with common tools, reusable software libraries, and centralized coordination. The planning and development of easy-to-use systems that delight the customer can be achieved if everyone involved puts a high priority on system usability, and acts accordingly.

Acknowledgments

We would like to thank our colleagues who provided valuable input to this manuscript, especially Robert E. Opaluch for his expertise in usability engineering processes, and Judy Cabtor for valuable comments. We would also like to thank the developers and systems engineers of the AT&T Operations Systems Business Unit for working with us to achieve consistent and easy-to-use systems, by identifying common elements across systems, and by trying a variety of approaches to produce consistency.

* Macintosh is a registered trademark of Apple Computers, Microsoft Windows is a registered trademark of Microsoft Corporation,

VUE is a registered trademark of Hewlett Packard, OpenWindows is a registered trademark of Sun Corporation, and Microsoft Excel is a registered trademark of Microsoft Corporation.

References

1. J. P. Cunningham, C. D. Blewett, and J. S. Anderson, "Graphical interfaces for network operations and management," *AT&T Technical Journal*, Vol. 72, No. 3, May/June 1993, pp. 57-66.
2. J. Nielsen, *Usability Engineering*, Academic Press, 1993.
3. R. E. Opaluch and Y-C. Tsao, "Ten ways to improve usability engineering—Designing user interfaces for ease of use," *AT&T Technical Journal*, Vol. 72, No. 3, May/June 1993, pp. 75-88.
4. J. D. Gould, S. J. Boies, and C. Lewis, "Making usable, useful, productivity-enhancing computer applications," *Communications of the Association of Computing Machinery (ACM)*, Vol. 34, No. 1, January 1991, pp. 74-85.
5. R. P. Beck, S. R. Desai, D. R. Ryan, R. W. Tower, D. Q. Vroom, and L. Mayer Wood, "Architectures for large-scale reuse," *AT&T Technical Journal*, Vol. 71, No. 6, November/December, 1992, pp. 34-45.
6. "Ergonomic requirements for office work with visual display terminals - Part 10: Dialogue principles," ISO CD 9241-10.2.
7. D. J. Mayhew, *Principles and Guidelines in Software User Interface Design*, Prentice-Hall, 1992.
8. C. M. Brown, *Human-Computer Interface Design Guidelines*, Ablex, 1988.
9. P. H. Callahan, "Expert systems for AT&T switched network maintenance," *AT&T Technical Journal*, Vol. 67, No. 1, January/February 1988, pp. 93-103.
10. J. Nielsen (ed.), *Designing User Interfaces for International Use*, Elsevier, 1990.
11. Digital Equipment Corporation, *A Digital Guide to Developing International User Information*, Digital Equipment Corporation, 1991.
12. D. Taylor, *Global Software: Developing Applications for the International Market*, Springer-Verlag, 1993.

(Manuscript approved March 1994)

Nuray A. Aykin is a member of technical staff in the User Interface Architecture—Documentation and Systems Group at AT&T Bell Laboratories in Holmdel, New Jersey. She leads user interface subteams in defining graphical user interfaces standards for network management applications, and is involved with the user interface design of various Network Systems products. She joined the company in 1989. She has a B.S. degree in industrial engineering and an M.S. degree in operations research from the Middle East Technical University in Ankara, Turkey, and a Ph. D. in human factors engineering from the State University of New York at Buffalo.



James P. Cunningham is the technical manager of the User Interface Architecture—Documentation and Systems Group at AT&T Bell Laboratories in Holmdel, New Jersey, where he and his group are responsible for AT&T-wide technical planning in the areas of documentation, graphical user interfaces, and usability engineering methods. He joined AT&T in 1984 with a B.A. degree in psychology from Stanford University, Palo Alto, California, and a Ph.D. in mathematical psychology from the University of California at San Diego.



Joseph P. Rotella is a member of technical staff in the Software Products Platform Knowledge Domain at AT&T Bell Laboratories in Columbus, Ohio. He joined the company in 1986. He has a B.S. degree in computer science from Rensselaer Polytechnic Institute in Troy, New York.

