

Carnegie Mellon's Andrew:
The Evolving User Interface of the Messages Program

Christina Haas, Consultant for Interface Design and User Testing

Daniel Boyarski, Graphic Designer and Consultant for Interface Design

Nathaniel Borenstein, System Designer and author of the Messages Program

Information Technology Center

Carnegie Mellon University

Pittsburgh, PA 15213

(412) 268-6717

Introduction

The Andrew project at Carnegie Mellon University is a prototype computing environment for universities, developed by the Information Technology Center, a joint venture of IBM and CMU (Morris, et al, 1986; Morris, 1988). The Andrew system includes the Andrew File System, a large distributed network file system (Howard, 1988); the Andrew Toolkit, an object-oriented environment to support user interface software development (Palay, et al, 1988); and the Andrew Message System, a multi-media mail and bulletin board system which utilizes the Vice network (Borenstein, 1988; Rosenberg, et al, 1987).

Designing the user interface for a computer system such as Andrew is a complex task, involving a variety of issues, such as visual/verbal communication, cognitive psychology, system design, the written word, usability testing, etc. Contributing to the Andrew project over the last several years have been individuals with a range of different backgrounds and academic disciplines from Graphic Design to Computer Science, from Engineering to Rhetoric. These individuals bring different perspectives on how to design and produce the user interface for a system and the resulting interdisciplinary give-and-take has been crucial to the design and development of Andrew's User Interface.

The Andrew project is now five years old; this paper focuses on one aspect of Andrew--the

Message System--in which the user interface has continued to evolve in useful, innovative, and productive ways. We begin with a brief overview of the Andrew Message System as it exists today. Then we describe some of the improvements made to the Messages program during the past year by examining two strands of work which have contributed significantly to the evolving user interface of the Andrew Message System: 1) the contributions of graphic designers in analyzing issues of visual/verbal communication; and 2) contributions from human factors testing and usability analysis examining how actual users learn, understand, and use the Andrew Message System.

The Andrew Message System Today

The Andrew Message System (AMS) is a portable, distributed, multi-media, multi-interface system for reading and sending mail and bulletin board (bboard) messages. The AMS supports multi-media messages, which may include line drawings, hierarchical drawings, spreadsheets, raster images, animations, and equations. It is explicitly designed to support a huge database of messages and an enormous user community. At CMU, it services over 1600 bboards, including netnews, the Dow Jones information service broadtape, and bboards on which newspaper cartoons appear as raster images.

The system incorporates an automatic "white pages" for doing name lookups, including phonetic matching of misspelled names. In addition, the system supports a number of advanced features such as voting on multiple-choice questions, private bboards, shared mailboxes, and automatic classification of incoming mail messages. The server-based architecture makes it easy for client interfaces to be ported to or built on almost any computer. For instance, associates of the Center for Educational Computing in English have developed the Comments program, which allows teachers to comment on student papers electronically (Neuwirth, 1988). Currently, interfaces to the AMS run on IBM RTs, DEC MicroVaxes, Suns, IBM PC's, Macintoshes, and Vax UNIX and VMS timesharing systems. For more detailed descriptions of the Andrew Message System, see Borenstein, et al (1988) and Rosenberg et al (1988).

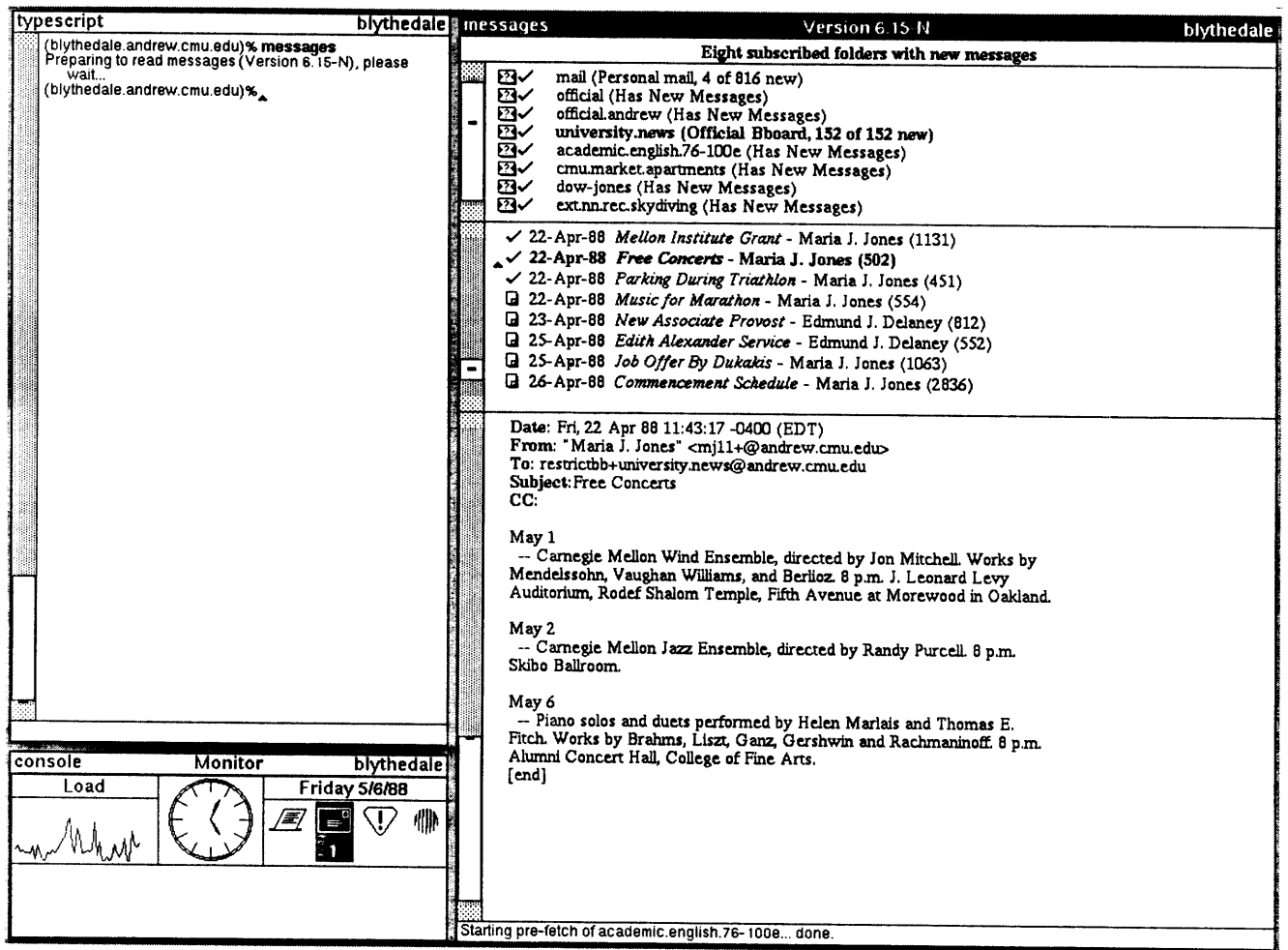


Figure 1. Reading Messages.

Thousands of students, faculty, and staff have been using the system during its development over the last few years for reading and sending personal mail, and for reading and posting to academic, technical, and general interest bulletin boards. Users are alerted--via a lit icon--when they have new personal mail and the mail and bboard facilities share a common interface. Figure 1 shows the Messages window, used for reading mail and bboards. There are three regions: the upper region, or folder region, which contains a list of subscribed "folders" (including a mail folder) which have new messages. The center region (or captions region) shows the titles, senders, and dates of the new messages in the selected folder (which is indicated as bold in the folders regions). The bottom region, or message region, shows the text of the message selected in the captions region. How the space within the messages window is allocated has been the focus of much discussion and several rounds of user testing and is discussed in more detail below.

Sending or replying to mail is done in a separate window (running under the same program) so that users may compose messages and view messages simultaneously. In Messages, users use the mouse, menus, scroll bars and other features much as they do in other Andrew applications. In SendMessage, text is entered, deleted, and moved as it is in the Andrew editors. Within SendMessage users can use general formatting features such as titles, special fonts, and indented text. In addition, the AMS is a multi-media mail system: users can insert drawings, tables, raster images and other insets. Figure 2 shows the Andrew screen of one user; the user is replying (in the lower window) to a message she has just read (in the upper window.) Note the mail icon in the Monitor window which indicates that the user has one piece of new mail and the drawing which the user has created within the sendmessage window.

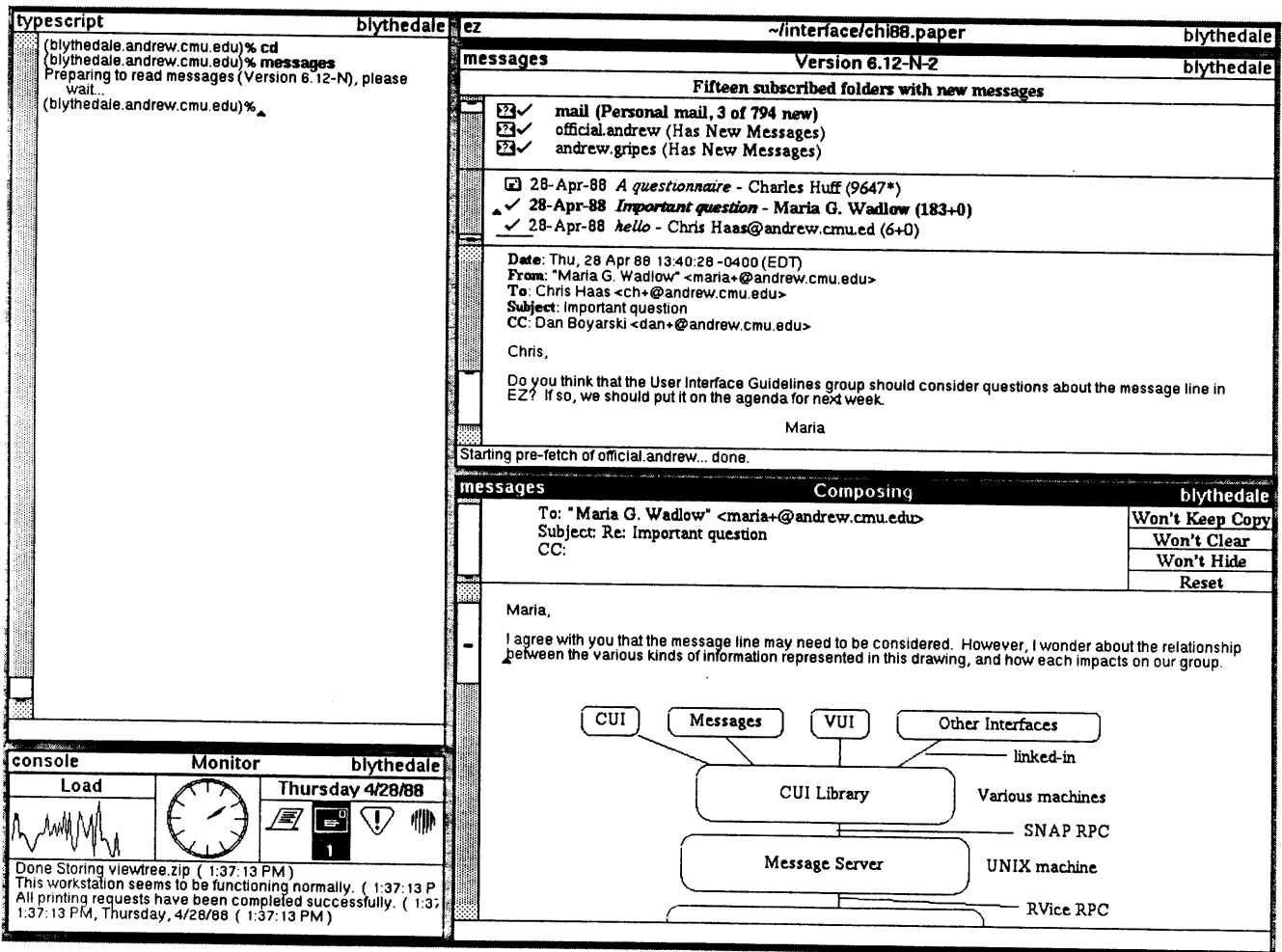


Figure 2. Reading and Sending Messages.

While the number of available bboards may number in the thousands, when users invoke the Messages program (by typing 'messages' in the typescript window or by using a pop-up menu) they typically see only a few bulletin boards: those to which they are subscribed which have new postings. Figure 3 shows a tree structure which symbolizes the configuration of Messages for one user, Pat. In addition to personal mail, Pat is subscribed to six bulletin boards, including **official.andrew**, a bulletin board containing official announcements such as printer down times and to which new users are automatically subscribed; **academic.english100e**, the class bulletin board for Pat's English class; and **ext.nn.rec.skydiving**, an external netnews bulletin board.

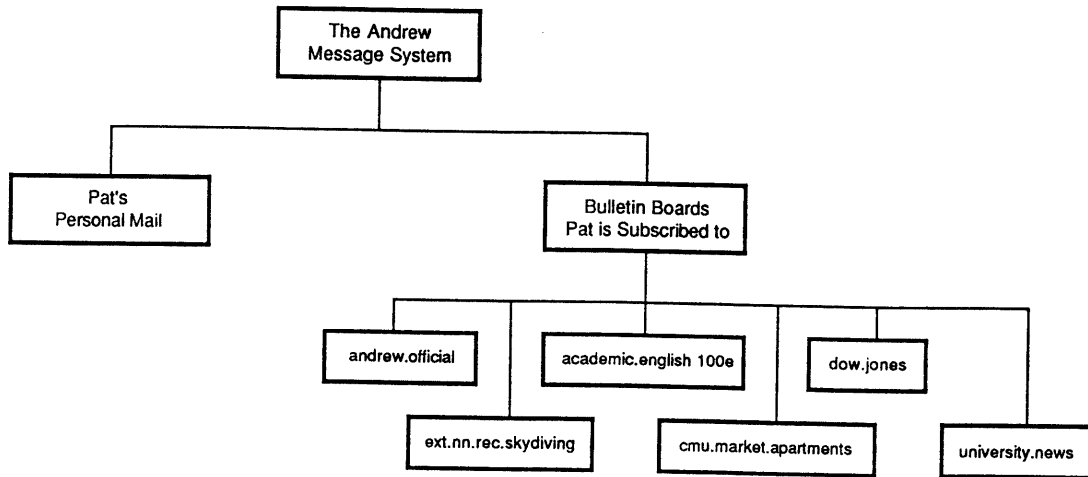


Figure 3. Pat's Messages

Improving the AMS Interface: The Graphic Designer's Perspective

Goals

When we first started discussing the physical appearance of Messages, we wrote down our goals:

Clarity in the presentation of information is our primary goal.

Clarity can be achieved a number of ways:

1. A logical hierarchy to the presentation of the information:
message **folders** first, message **captions** second, the message **text** third.
(*macro view to micro view*)
2. Consistency in the location of the information. Surprises are not always appreciated.
3. Consistency in the size of window sections.
4. Readable typography. This refers to the arrangement of the type/text within each of the sections; it also refers to the typeface (font) and type sizes in use, line length, and line spacing.

A second goal is to allow the user to read, reply, post, and organize one's messages *easily* and *efficiently*. It has been proven that when a user has a positive experience running a program, he/she will gladly return to this program the next time a similar task needs to be done.

Icons

The task was to show a distinction between mail and bboard postings in the middle section of the messages window, where the message captions are displayed. Icons were considered, since they often function as a visual shorthand for words. The design process initially involved :

1. understanding the content of each icon: what is it to say?
2. understanding the location of each icon: where will it appear, and how often; will it be in the company of other icons or text?
3. understanding other constraints: how large can the icon be; does it have to line up with type/text or other icons; how large of a family of icons will this involve?

The Mail Icon

Several ideas were considered, including an envelope and a traditional "rural" mailbox. This mailbox, we argued, was not appropriate, given the fact that not many people today own such a mailbox, while others have never even *seen* such a mailbox. Today's apartment dwellers are more familiar with wall mail slots, which are opened and locked with a key. Therefore, the image of the "rural" mailbox, mounted on a post, with its little flag, is meaningless to a larger and larger segment of our population. (The same argument can be made about an hourglass to indicate the passing of time.)

The image of an envelope with a stamp and an address became our recommendation. This is a familiar image, one that suggests new mail; a positive connotation. We preferred this to the back of an envelope because we consider the back of an envelope to be a rather passive image (waiting for some action to be taken), while the front of an envelope with stamp and address is a more active one.

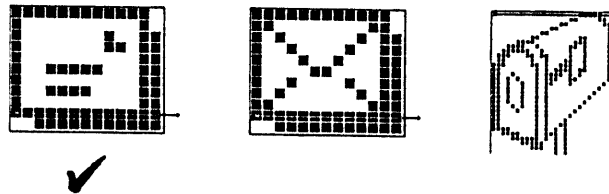


Figure 4. Mail Icon Options

The Bboard Icon

In contrast to the envelope for the mail icon (a horizontal rectangle), the icon for a bulletin board message became a vertical piece of paper with a folded corner (now part of our visual vocabulary for display graphics). This icon was fairly easy to design, given the visual context of a bulletin board with multiple pieces of paper tacked up on it. Our campus is filled with such bulletin boards, not unlike other campuses and other environments where people need to communicate with one another.

Various iterations on this icon explored the visual suggestion of tacks at the page's corners (too face-like), and the suggestion of text on the page (too busy). The directness and simplicity of the single page was clear enough.

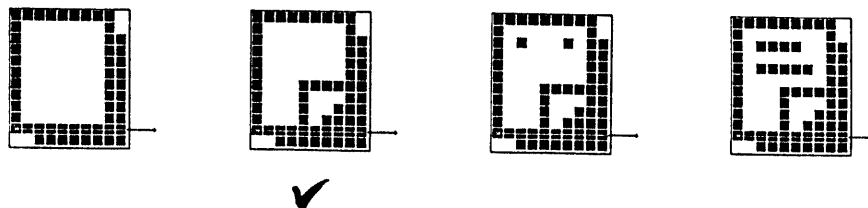


Figure 5. Bboard Icon Options

The Other Icons

After the two basic message icons were accepted, we proceeded with expanding the family of icons. Without going into detail, here are the other icons and their functions:

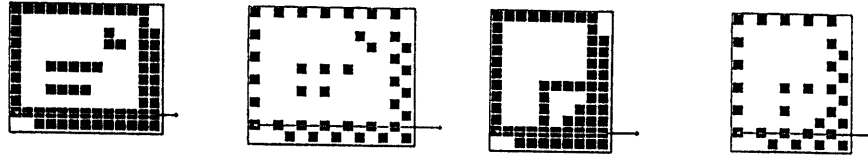


Figure 6. Icons: New Mail, Deleted Mail, New Bboard Posting, Deleted Bboard Posting

The "check mark" is an icon that has been around the ITC for many years and has proven its value as the iconic indicator that says, this message has been read. There was no reason for me to improve on it.

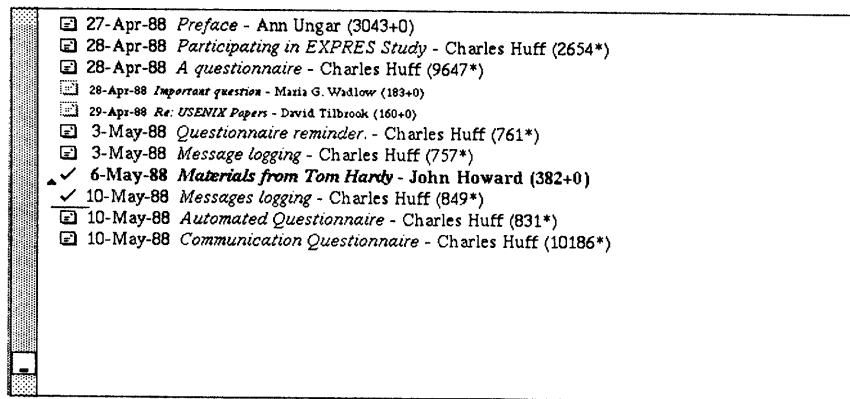


Figure 7. Middle Section of Messages Window with Icons

Typography

Along with the discussion on allocating screen space within the Messages window (see later section of this paper), was the task of making recommendations on typefaces (fonts), sizes, and weights, as well as the arrangement of this type in the various window sections.

The typefaces available within the Andrew system are a serif font (like Times Roman), a sans serif font (like Helvetica), and a fixed-width typewriter font (like Courier). For legibility reasons, we selected the serif font for all the information display in messages. Weight and size changes within the serif family, along with formatting options, gave me enough signalling devices for a clear hierarchical structure of the information.

As Messages evolved, we had to consider ways to indicate:
a message that has not been read
a message that has been read
a message that has been deleted

A Message That Has Not Been Read

Recommendation: All message captions are listed in one typeface (serif) and a single weight (normal), with italics used to call out the message title. The caption line includes four pieces of information: the **date** the message was posted, the message **title**, the **author**, and the number of **words** in the body of the message. The message title is *italicized*, to visually separate it from the rest of the information. It is given added emphasis with a character space between it and the date, and a flush left arrangement, thereby giving the user easy visual access. The goal here was to give it visibility and accessibility, and not to have it appear buried in the grayness of the text.

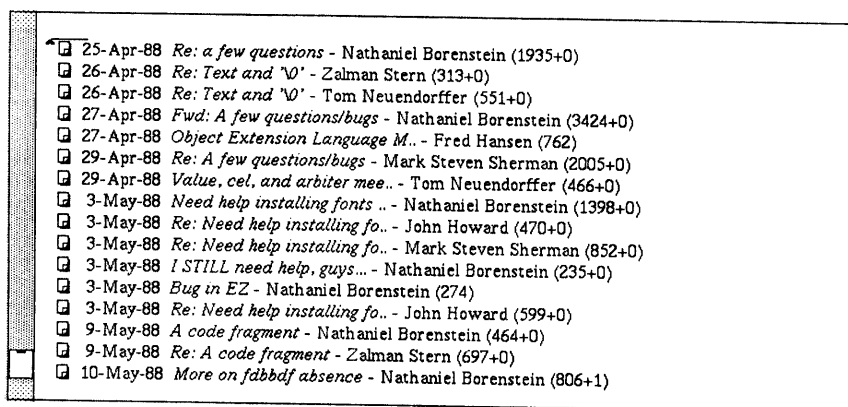


Figure 8. Messages that Have Not Been Read

Our recommendation involved the combined communicative potential of words and icons. As shown above, the icons are used as scanning devices, then as bits of information (new, has been read, deleted) and as pointers to the message captions which sit next to them on the same baseline.

The middle window section employs several margins (tabs) for a clear, hierarchical structure. The first margin contains the icons, the second one contains the date, and the third margin contains the message title, its author, and the number of words in the message itself. We tried a new margin for each bit of information, but the various lengths in the titles and the authors' names proved to be a problem. It became difficult to read across a rather long horizontal line that was peppered with short and long bits of type and five margins.

A Message That Is Selected and Read

To select a particular caption, the cursor (a left pointing arrow) is pointed to that caption and a mouse button is clicked. The result is that the entire line of type turns **bold**. The body of the message appears in the bottom section of the window. This rather obvious, but very clear signal, is most effective in the midst of many titles. The single bold line stands out clearly

against the field of grey type. This signal is used *consistently* throughout the Messages program, and the Andrew system. When a new caption is selected, the old caption returns to its original normal weight, while the newly selected one turns bold. Accordingly, the icon changes from the mail or board icon to a check mark icon, signifying that the user has read that message. The physical change from the envelope or page icon to check mark icon is a clear visual signal that a change in status has occurred.

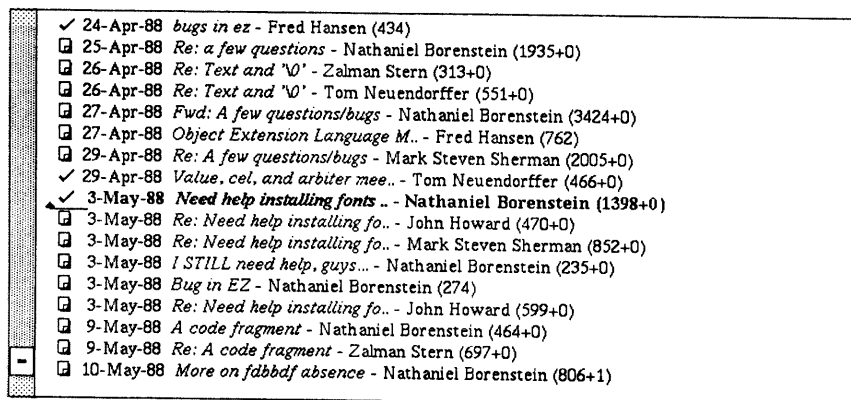


Figure 9. A Message that is Selected and Read

A Message That Is Deleted

Deleting a message is accomplished with a menu item selection. When this is done, the caption line changes in appearance: it shrinks to a smaller point size. This was actually the recommendation of Nathaniel Borenstein, the creator of Messages. Coupled with that change in appearance, the icon changes to a faded envelope or page. Seen in the context of Messages, these changes are clear signals to the user.

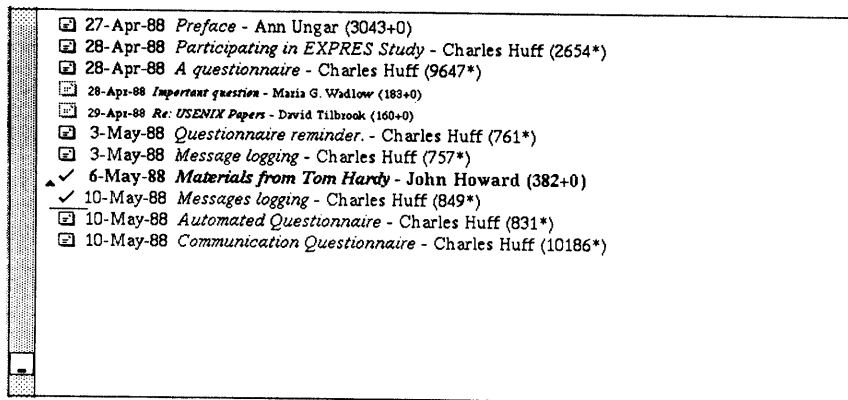


Figure 10. Several Messages: Unread, Selected, Deleted

Generally

Observing ourselves and other people using Messages, we are reminded that one can move around rapidly, selecting, reading, deleting, moving on. Therefore, the visual cues and signals have to be visible, recognizable, and clear. Icons and typographic cues that make intuitive sense are easily remembered by new and/or occasional users. They allow the user to concentrate on the task at hand, and get through the task as efficiently and effortlessly as possible.

Improving the AMS Interface: User Testing

The AMS interface has also benefitted from testing the system with real users. Not only is the system used by a group of sophisticated and vocal users at the Information Technology Center, but several iterations of user testing have been conducted with users from the campus community with varying degrees of expertise. We have conducted two kinds of user tests: 1) general, exploratory studies to determine how people actually use the Andrew Message System, and 2) more focused, issue-driven experiments designed to answer specific questions about the interface. Each of these strands of testing has been important and fruitful: the exploratory testing has allowed us to learn from users and often to be "surprised" by how they use the Andrew Message System in a way that wouldn't be possible with more controlled studies; the issue-driven studies have allowed us to address specific questions about problematic features of the Andrew Message System. Often the exploratory studies have unearthed questions which we examined more carefully with controlled studies. Below we give examples of each of these strands of user testing that have contributed to the user interface of the Andrew Message System.

Exploratory Testing

The purpose of the exploratory testing is to get more information: about who uses Messages, how often, and for what purposes; about problems or concerns that users have; and about users' opinions of the Andrew Message System. We have used a variety of methods to gather this information: we have walked about public workstation clusters on campus and observed if, how, and how much users seem to be using the Andrew Message System. We have also conducted interviews with users, some of a series of informal questions we asked users as we observed them. Other interviews have been more structured and planned. We ask user such questions as how often they send and receive mail, how many and which bboards they are subscribed to, or what improvements would make Messages easier to use.

Another example of exploratory testing is a series of surveys (Haas & Zalevsky, 1988; Rago & Haas, 1987) which have been distributed at the end of each semester to the instructors who teach new students on campus about Andrew. The Computer Skills Workshop, offered to incoming students at Carnegie Mellon, includes a six week introduction to the Andrew System. Each semester we submit a survey to the teaching assistants who instruct the course. The primary goal of the survey is to collect data about the level of Andrew mastery of the CSW students. The bulk of the survey is a series of Likert-scale questions which ask instructors to indicate the number of students in their classes who have achieved mastery with various Andrew features; instructors chose from these responses: *almost all*, *more than half*, *less than half*, or *very few*. In addition, we solicit opinions from the instructors about Andrew. The survey asks Likert-scale and open-ended opinion questions about five aspects of the Andrew system, including the Andrew Message System.

The results of the most recent survey (conducted in October 1987) showed impressive levels of use and satisfaction for the Andrew Message System. Virtually all of the instructors surveyed indicated that *more than half* of their students were facile and comfortable with reading and sending mail and reading bulletin boards; almost 88% of the instructors indicated that *almost all* of their students had reached this level of competence after six weeks of instruction. Further, in an open-ended question on the survey which asked instructors to indicate Andrew features or programs which they personally found useful or valuable, two-thirds volunteered the information that Messages was one of these useful programs. This level of satisfaction was higher for Messages than for any other Andrew feature or program. And finally--the most heartening result for those of us who have worked together to improve the Andrew Message System--when instructors were asked to list any positive changes or improvements that they had noticed in Andrew in the last year, three-quarters volunteered that they had noticed actual improvements in the Andrew Message System.

While these results were encouraging, the survey also pointed out a potential problem for new users in the Andrew Message System. We noticed that for the second year in a row instructors indicated that students were less facile with posting to bulletin boards than they were with other Andrew Message System functions. Specifically, 30% of the instructors indicated that *less than half* of their students were able and comfortable posting messages to campus bulletin boards. In other words, the level of competence with **posting to bulletin boards** was substantially lower than with the other Andrew Message System functions, **reading mail**, **sending mail**, and **reading bulletin boards**. We are currently conducting follow-up studies to find out more about this puzzling result.

Issue-Driven Testing

In contrast to the exploratory testing, issue-driven testing is usually designed to answer specific questions about features within the Andrew Message System. For example, developers of the

Andrew Message System were interested in the most beneficial way to allocate screen space within the Messages window. Specifically, two rival Message window layouts had been developed which allocated space to the regions within Messages differently and arranged the regions within the window in two different ways. These two rival interfaces were generally referred to as the “stacked” interface (pictured in Figure 3), in which the three regions are arranged in a stack; and the “next-to” interface (Figure 4), in which the folders region was next to the captions region (the message region remained in the same location). We designed a series of studies which were driven by the question: “What is the more useful way to allocate and arrange space within the Andrew Message System window?” (Zalevsky, 1987).

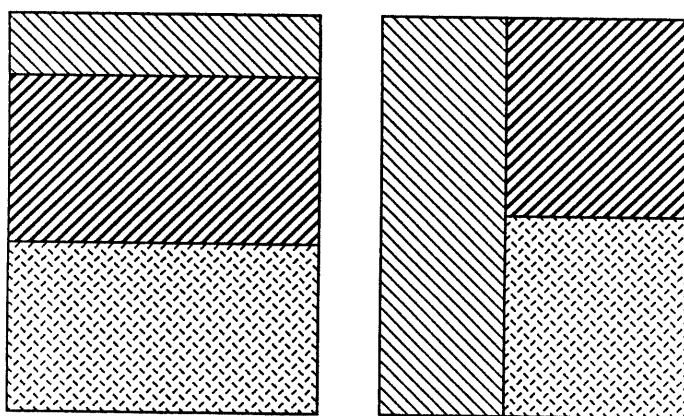


Figure 11. “Rival” layouts for three regions in Messages.

Users new to the Andrew System were introduced to and trained in use of the Andrew Message System. Users were introduced to both of the rival layout schemes, the “stacked” interface and the “next-to” interface. The order in which users were introduced to the contrasting layout schemes was counter-balanced across subjects. On a subsequent day users performed a recall task: they were asked to role-play teaching the Andrew Message System to another new user of the system. The researcher took notes and tape recorded subjects’ recalls. Following this recall task, we interviewed the subjects about which layout scheme they preferred and why. Results of both the recall task and the discussions with users revealed that the only effects were for order: subjects both preferred and recalled better the layout scheme they had learned first. That is, regardless of which layout scheme subjects learned first, that is the layout scheme they liked and remembered best.

Developers used the information obtained in this study to determine that the layout scheme of the Messages window might best be determined by individual users. The “stacked” interface was chosen as the default for new users, but--by simply choosing from a menu of display options--users can change the layout they see to the “next-to” interface. One of the most important results of this study is that it underscores the importance of consistency within an interface. Bearing in mind that users seem to become accustomed to and prefer system features with which they are familiar is at least as important as determining specific interface features.

Conclusion

Interface design is crucial to the success of any computer system since, to the user, the interface *is* the system. This design process, however, is as complex as it is important. The user interface of the Andrew Message System has benefited from the expertise of a range of people with different background, training, and contributions, including graphic designers and human factors evaluators trained in cognitive psychology, education, and document design. With help from experts in a variety of domains, the AMS has become an impressive and widely used showpiece of the Andrew system.

Acknowledgments

Responsible for the development of the AMS was the Message System Group at the Information Technology Center: Jon Rosenberg, Nathaniel Borenstein, Craig Everhart, and Adam Stoller. Tracy Futhey, Director of Computing for the College of Humanities and Social Sciences (and John Stuckey in that position before her), supported collection of survey data from the instructors of the Computing Skills Workshop. Monica Zalevsky and Vincent Rago assisted in the collection and analysis of the data from the surveys. Current ITC Director Alfred Spector and former ITC Director Jim Morris deserve thanks, as do members of the User Interface Group: Andy Palay and Maria Wadlow.

References

- Borenstein, N., Everhart, C., Rosenberg, J., & Stoller, A. (1988) A multi-media message system for Andrew. *Proceedings of the USENIX Technical Conference*, February.
- Haas, C. & Zalevsky, M. (1988). *Report on Survey of Computer SKills Workshop Teaching Assistants*. Information Technology Center Occasional Paper.
- Howard, J. H. (1988) An overview of the Andrew File System. *Proceedings of the USENIX Technical Conference*, February.
- Morris, J., Satyarayanan, M., Conner, M., Howard, J. H., Rosenthal, D., & Smith, F. D. (1986). Andrew: A distributed personal computing environment. *Communications of the ACM*, 29 (3), 184-201.
- Neuwirth, C. (1988) *Comments: Computer Support for Response to Writing*. CECE Technical Report #4, Carnegie Mellon University.
- Palay, A. J., Hansen, W. J., Kazar, M. L., Sherman, M., Wadlow, M., Neuendorffer, T. P., Stern, Z., Bader, M., Peters, T. (1988) The Andrew Toolkit: An overview. *Proceedings of the USENIX Technical Conference*, February.
- Rago, V. & Haas, C. (1987) *Report on Survey of Computer SKills Workshop Instructors*. Information Technology Center Occasional Paper.
- Rosenberg, J., Borenstein, N., Everhart, C., & Stoller, A. (1987) An overview of the Andrew Message System. *Proceedings of the SIGCOMM '87 Workshop, Frontiers in Computer Communications Technology*, August.