Chapter #

Voice Messaging User Interface

Harry E. Blanchard & Steven H. Lewis
AT&T Labs

Key words: voice mail voice messaging unified messaging universal mailbox

Abstract:

Voice mail is certainly one of the most common interactive voice response applications in telephony. And, when we talk of designing these systems to be easy to use, there is much overlap with general IVR system design, but there are also significant special characteristics to voice mail systems. This chapter discusses designing voice mail systems for usability with particular emphasis on the special characteristics, demands, conventions, and standards associated with voice mail. In addition to coverage of the ubiquitous touch-tone user interfaces, this chapter also discusses the state of the art in voice mail user interfaces: novel prompt and menu structures, the use of automatic speech recognition, unified messaging ("universal mailbox"), and multimedia mail.

1. INTRODUCTION

Voice messaging, or voice mail, is a computer application integrated with the voice telephone network that allows people to receive, send, and store recorded audio messages. From modest beginnings in the early 1980s as a replacement for answering machines in business environments, voice messaging has grown to become an essential business tool and an increasingly common consumer service. By the end of 1998, a good estimate of the popularity of voice messaging can be seen in the U.S. number of

mailboxes, with about 60 million business "seats" with mailboxes, an additional 12 million or so residence lines with call answering mailboxes, and some 6 million or so mailboxes behind cell phones and pagers. Numbers for the rest of the world are harder to come by, but the total international market for voice messaging has been estimated by some of the equipment makers as about as large as the U.S. market. Millions of calls each day are answered by voice messaging applications and retrieved by mailbox subscribers.

The typical voice messaging application is controlled by users entirely from a conventional touch-tone telephone. Thus, voice-messaging applications fall into the class of Interactive Voice Response (IVR) services, most are touch-tone user interfaces (see Chapter 7). As many have noted, the touch-tone, or DTMF, user interface is impoverished, both in the user input, which is limited to the 12 keys on the standard touch-tone telephone, and in the system output, which places large processing demands on users because of its serial audio nature.

It is useful to think of voice messaging as services built from a set of messaging capabilities. These include

1. *Call answering:* When the user is doesn't answer a phone call or is busy on another call, the voice mail system answers the call and can take a message. This function is parallel to the capability provided by a telephone answering machine, except that the voice mail system can take messages while the user is talking on their telephone and is able to provide the caller with various options.

2. *Message retrieval and management:* As with answering machines, voice mail allows users to play messages left for them, and it allows them to store those messages and play them back later.

3. *Sending messages:* Voice mail systems can provide a functional analog to e-mail that an answering machine cannot: the ability to record a messages in the voice mail application and send them directly to other user's mailboxes on the same or connected systems. Sending can be invoked by subscribers who originate messages or who reply or forward to messages they receive.

4. *Call delivery:* Another form of voice messaging intercepts calls in the network that are not answered or busy and offers the caller a chance to record a message to be delivered later. The system then keeps trying the

line and delivers the message to a person or an answering machine when the line becomes free.

5. *Bulletin boards*: In this form of voice messaging, mailboxes do not belong to individual subscribers but rather are used by many people to leave and retrieve semi-public messages. This form of voice messaging is not common in the U.S.

Most business voice messaging applications provide the first three of these capabilities to subscribers using the public telephone network or private (PBX) telephone systems. Residential mailboxes typically enable call answering and message retrieval; sending messages is usually not available until there is a sufficient subscriber base. Call delivery is a form of voice messaging that doesn't assume a mailbox. Increasingly in new services, combinations of these functions have been embedded as the messaging one component of even more feature-rich systems that provide call management and intelligent agent capabilities.

Voice mail systems provide a special challenge to the user interface designer beyond the challenges of designing other IVR systems. Voice mail is distinct from call-direction and informational IVR systems in several respects:

- Voice mail systems are highly complex and feature-rich, and it is difficult to reduce the complexity of the system by not providing these features. At any one point or menu in the user-system dialog, the number of choices the user has available may outstrip the dozen keys available for the user to make a response. This is particularly true in dealing with a message, both while listening to it (when the "menu" of choices is implicit) and immediately afterwards. The common strategy of breaking up functions into a series of hierarchical menus cannot always be done, e.g. in responding to a message, many functions must be available immediately after listening to the message.

- Unlike most IVR systems, voice mail systems are used day in and day out, often several times a day. Many users are highly practiced and need to accomplish transactions quickly and efficiently. Nevertheless, there are also novice users, who are faced with a necessarily complex, feature-rich user interface to master.

- In addition to the usual prompted menus common to IVR applications, voice-messaging systems need to allow users to navigate around the

system freely, and operate on voice and DTMF data. This often leads to more use of common commands, often invoked with multiple, sequential keypresses.

- A voice mailbox has two distinct sets of users. There is the mailbox box owner, often called the *subscriber*, who gets and manages messages everyday. Then there is the *caller* who leaves a message for the mailbox owner using call answering. Many callers are *not* subscribers to the particular voice mail system they must interact with to leave a message. Callers come into the call answering user interface as the ultimate novices, with a set of expectations based upon the conventions of making a phone call and using answering machines and a variety of voice mail systems from different vendors which they encounter. Attention to these user expectations and the encouragement of cross-application standards are particularly important.

- Standards and consistency of design are highly significant to voice mail use. Besides the needs of callers leaving a message, described above, subscribers are also facing needs for consistency. There are an increasing number of users of voice mail who have one mailbox for their work phone and another mailbox for the home telephone. And, because of the development of the telephone industry, the office and home voice mail systems are almost always from different manufacturers, with different user interfaces and different sets of functions.

Over the last few years, we have been involved in the design and competitive analysis of residential, business, and cellular voice messaging user interfaces. In this chapter we will discuss voice mail user interface design with a focus on design decisions which are significant to voice mail systems. Readers should also be familiar with the general principles of IVR and touch-tone system design, as presented by Gardner-Bonneau and Schwab in this book.

2. THE TOUCH-TONE VOICE MAIL USER INTERFACE

The touch-tone user interface for creating and getting messages will be the most familiar to the business user and the dominant interface for messaging for the foreseeable future. The touch-tone user interface is series of *transactions* where the system provides audio instructions to the user who

directs the system by pressing a key or sequence of keys on the touch-tone, or DTMF, telephone keypad or by recording speech.

The dominant metaphor for voice mail has evolved. Early voice mail systems used an answering machine metaphor in the design and presentation of the user interface. At the time, most answering machines used magnetic audiotape for recording and reflected a tape recorder metaphor. Early voice mail systems used tape recorder terminology, offering options such as "rewinding" or "erasing" voice messages. Current voice mail systems downplay the tape answering machine metaphor, even as answering machines themselves are increasingly likely to use digital, tapeless recording technology. Today, the parallels with e-mail are more compelling, so terms such as "delete" and "forward," familiar to e-mail users, are becoming more common in voice mail. The e-mail metaphor for voice mail is likely to persist, considering the burgeoning use of Internet e-mail by a wide and diverse population of users and the convergence of voice mail and e-mail into unified messaging and universal mailboxes.

2.1 Common Elements of Touch-tone Transactions

Touch-tone user interfaces can be usefully analyzed using the general user interface categories of system output, user input, and user-system dialog. For touch-tone user interfaces, system output is made up of announcements, including menus and prompts; user input is single or sequential DTMF keypresses, voice recordings, or no action; and dialog includes interactions between input and output, including timeouts and error handling. From a task analysis perspective, the caller's focus in on leaving messages, which can be done by simply speaking and then hanging up, but can also include some manipulation of the voice input. The subscriber's focus is on retrieving and dealing with messages, as well as sending them.

Several good summaries of the general principles for designing touch-tone IVR dialogs are available (Halstead-Nusslach, 1989; ISO/IEC 13714, 1995; Marics and Englebeck, 1998; Schumacher, Hardzinski, & Schwartz, 1995; Schwab, this book). Although voice messaging differs from generic touch-tone IVR applications in many important ways, our experience has been that generic touch-tone design guidelines work.

2.1.1 Prompts

Voice messaging application trees can be "deep" and "broad," but at most points in the dialog people usually have a good idea of what they want to do. With many menu options, the convention of first stating the action then the required key to perform that action (e.g., To do X, press Y) applies. For example,

> *To get your messages, press 1*
> *To send a message, press 2,*
> *To record or change your greeting, press 3.*

The action-response order of each of the three prompts in this menu reduces the mental work needed. The action-response order is consistent with what Clark and Haviland (1977) called the "given-new contract" in conversation and has leads to faster responses and fewer errors (Englebeck and Roberts, 1990).

However, there are cases where the response-action phrasing sounds better; exceptions to this general rule may apply when there are only one or two options, e.g.,

> *Press 1 for yes or 2 for no please.*.

One prompt strategy that balances the needs of experienced users with those of novices is to use staged prompts, where an initial terse prompt is followed by a short embedded pause and then a more extended prompt (Brems, Rabin, & Waggett, 1995; Leopold & van Nes, 1984). Thus, for example, the prompt for addressing a message might be:

> *To what number? …[2 seconds of silence]… Enter a mailbox number followed by the pound key.*

2.1.2 Interruptibility

New users are likely to wait until all the options are played before selecting a key to choose an option in a menu. However, voice messaging applications wouldn't be tolerated by experienced users unless they allowed menu-selection keypresses before the end of entire menu prompt, and indeed, also allowed users to provide keypresses that apply several menus or systems states ahead of the playing prompt. Indeed many experienced

voicemail subscribers get messages by giving their ID, their password, whatever delimiters are needed, and the main menu key that starts messages playing as if it were one long DTMF string

The capability to press a key before the menu has finished playing and have the system immediately stop playing the prompt or announcement and act on this input is called *dial-though*. Unlike an IVR system that might be used once or occasionally, dial-through is absolutely essential to voice mail users. For example,

> System: *To get your*
> User interrupts: 1
> System: *First message…*

Dial-ahead, which is the entry of a sequence of commands before the prompt(s) have been played by the system, is also essential. When voice mail designers have chosen to disable dial-through and dial-ahead, for example to let users know if they have system or priority messages or are running low on storage, users are likely to complain.

2.1.3 Time-outs and Re-prompts

When no response is detected from the user after some time-out period, the system should respond. Systems have two major time-outs: the *time-out for no action* and the *inter-digit time-out.* The time-out for no action is the maximum time the system waits for any response after a prompt. If this time-out is too long, the user often doesn't know what is happening; if it is too short, the user can feel rushed. In our experience, a time-out for no action of 5-6 seconds works well. The inter-digit time-out is the maximum time the system should wait between keypresses when more than one digit is expected, for example, when the user is entering a telephone number. In regular telephone dialing, this time-out is 15 seconds. In voice messaging and other IVR applications, our experience has been that a time-out of 5-6 seconds works here as well.

A time-out for no action should almost always result in a re-prompt for user action. The re-prompt should rephrase the focal options and may present all of the options available, including hidden or fall back options (such as returning to the main menu). A better re-prompt might orient users by telling them where they are in the menu hierarchy or what task they are doing. For example,

[User times out to menu following message.]
System: *Your are listening to a new message.*
To replay the message, press 1 …

After repeated time-outs of this sort (typically three is a good number) the system should do something other than re-prompt. First, it should inform the user that a special action is going to occur, and then perform that action, such as disconnecting or connecting to an attendant.

What the system should do when an interdigit time-out expires depends upon the context in which it happens and the system capabilities. It is almost always useful to interpret the time-out as a delimiter. The system then should attempt to process the data, accept it if it can be interpreted as valid, or prompt for more information, or for re-entry, depending upon the context and sophistication of the software.

So, for example, the user may enter seven digits when a telephone number is requested and time-out. The system may then interpret the number by assuming the three-digit area code, or prompt the user with

You must enter a 10-digit number beginning with the area code.

Two other time-outs are a time-out when the user is providing speech input and the system has detected a period of silence, and a time-out after the system has prompted for touch-tone string input, such as an address or a user ID, and no initial digit has been provided. While these are conceptually distinct types, in our experience the 5-6 second time-out works in these cases as well.

2.1.4 Feedback

Any user action or response must result in feedback from the system. Feedback to valid input is usually the progression of the system to the next prompt, menu, or message. Feedback does not mean that the requested operation is completed. If it is not, the system should inform the user after three seconds of no response that their request is still being processed. The cases which challenge timely feedback in systems usually involve extensive database access, which in voice mail systems may be address lookups.

2.1.5 Feedback to Errors

When a user provides incorrect input, as defined by the system, the system should inform the user of the error in a polite fashion that does not blame the user. Words such as *"Invalid entry"* are abrupt and not very informative. As with re-prompts to time-outs, the system should orient users by reminding them of what menu they are listening to or what task they are doing, followed by a full list of options that are available for example,

> *Five doesn't work here. You are in the main menu. To get your messages, press 1. To send a message, press 2. …*

2.1.6 Menu Length

A generally accepted principle of touch-tone design is to limit the number of options in a menu to approximately four (Englebeck & Roberts, 1990). Due to the feature richness of voice mail, fitting four options to a menu can be a challenge or an impossibility. For example, in the menu which is offered to the user after listening to a message, the user has to be given all the options for dealing with the message including deleting the message, replying to the message, forwarding the message, saving and/or skipping the message, and replaying the message. Often, other options will also apply. Splitting these functions at this point can itself be a serious usability problem, so, despite the complexity of the menu, it should offer all the message disposition options. In the end, the classical consideration for trade-off between depth and breath of menus applies (e.g. Lee & MacGregor, 1985; Norman, 1991).

2.1.7 Mapping of Keys to Options

Voice mail menus should follow the principles used for design of any touch-tone menu system. Key mappings are driven by a small set of guidelines that make good sense individually but can lead to conflicts when taken as a set.

Generally, the most frequently used or default action should be assigned to the 1 key, the next most frequent to 2, and so on (Marics and Englebeck, 1998). An exception to this rule concerns conventions or standards for key assignments. In particular, if a prompt requests a yes/no answer, the 1 key should be "yes" and the 2 key should be "no."

Menu options should be numbered, if possible, without gaps in the ordering. That is, options in a menu should be numbered consecutively from 1 to 9, the first option is 1, the second option is 2, etc. This aligns with user expectations. If a gap appears, users may overlook the renumbering, or think they missed an option while they were listening. If options are presented out of order, then users may lose track of the key-action mappings.

This rule always involves a trade-off with other factors. For example, a message disposition menu appears after the user listens to a message. Say, for example, option 5 in the menu forwards a message. If the user receives a "private" message (a "private" message, in the parlance of most voice mail feature descriptions, is a message which cannot be forwarded to third person), option 5 should not be offered. Rather than moving option 6 to option 5 in the message disposition menu for a private message, it makes more sense simply to skip option 5 for these messages (i.e., to offer option 4, then offer option 6), in order to keep the canonical assignments the user learns for this one menu. It is probably particularly important that the mappings of keys for "destructive" or irreversible actions, such as erasing things or canceling actions, be consistent across menus.

2.1.8 Global Commands

Global commands are keypresses that are common to any menu in the user dialog, and may be used to invoke the same or similar features. One advantage of defining global commands is that they provide users with a tactic to universally "escape" out of situations in which they are confused, e.g., by going to a main menu or getting help. ISO/IEC 13714 specifies a "control menu" which is a set of options that are active at all points in the voice mail user interface. Control menu options, as defined by the standard, include:

- Going to the top level menu
- Selecting a language
- Force system to disconnect
- Cancel an action or backup to previous menu
- Invoke on-line help announcement
- Return to prior context

The advantage of specifying global commands or a control menu is that users can learn a specific key sequence for a common activity that need not be prompted.

2.1.9 Use of the "#" and "*" Keys

The "#" key is perhaps the key with the most fixed conventional meaning of any of the touch-tone keys, and has widely-accepted conventions of usage. The "#" key is used to indicate delimiting or completion, and moving ahead in the system. The "#" key should be used to indicate the end of a variable-length string of numbers, e.g., the end of a telephone numbers or credit card numbers. The "#" key should be used to indicate the end of voice input, i.e., to indicate the end of a recording. It is also used to skip to the next message in a queue and by callers to skip the subscriber's outgoing call answering greeting and go directly to the record tone. If "#" is used in other circumstances, it is recommended that it be used to indicate completing of an on-going action or entry.

Two common conventions appear for the "*" key. One convention reserves the key entirely for canceling user input or task steps or for moving back to the previous logical option or choice in the call flow. The other convention uses the key as a way to access a further set of menu options beyond the one-nine single-digit options.

Voice mail systems often have a large number of options, and voice mail often makes use of global options that pertain to all menus, in other words, a "control menu." Because of these features, it makes more sense to choose the convention which allows the "*" key to access more options, rather than restrict it to a generalized back-up/cancel/delete function. This use of the "*" as an escape or command key is defined in a recent ISO/ANSI voice messaging user interface standard (see the section on standards, below).

2.1.10 Unprompted Options

Although there is a general guideline that claims that no available option should go unprompted, there may be some circumstances where it is reasonable:

- Global keys. These are common global options that are not significant for a transaction, and can be expected to be well known. For example,

the system might refrain from prompts for "*0" for help on every menu in the user interface. Instead, prompt only when "*" is pressed, or provide an announcement at the beginning of the user dialog such as "For help at any time, press "*0.""

- Hidden treasures. Key sequences from a previous version or other vendor application provided to cater to user habits. When the application has its own key sequence to accomplish an action, e.g., "3" to delete, but users may know another key sequence from a previous version, e.g. "*3" to delete, the designer may want to allow "*3" to complete the action instead of returning an error message. However, it may be wise not to prompt for "*3" to delete, so new users don't learn obsolete keys. (This is analogous to Microsoft Excel accepting Lotus 123 key commands in its formulas, even though it has its own formula syntax.)

2.2 Call Answering

Because people bring with them rich expectations from their familiar interaction with answering machines, the designer flirts with disaster to violate the well-familiar script of answering machines:

1. recorded message in the called party's voice, saying "leave a message"
2. a record tone
3. caller leaves a message
4. hang-up terminates and saves message for subscriber

The voice mail system, however, provides a rich set of useful functions for the caller. Editing functions allow callers to delete and re-record a message they don't like, to abandon leaving a message (without leaving a blank or half-recording message after hang-up), and to navigate elsewhere in the voice mail system (e.g., leave a message for someone else in the same office, or go to the caller's own mailbox and get their own messages).

Figure 1 depicts a very common call answering call-flow which preserves the answering machine style of interaction while opening up the editing features. The caller who hangs up after leaving a message is confident in knowing that a message is left for the person called. The principle that hanging up should result in accepting a message is very strong—also based upon answering machine behavior—so strong, in fact, that it should, perhaps, be considered as the default action whenever the user is making a

recording. Certainly, when recording a message to be sent by voice mail, and even when the user is recording a greeting, hanging up should trigger accepting and saving the recording.

The caller who enters a keypress is greeted with editing options. The "#" key is required by ISO/IEC 13714 as the delimiter for voice input. The "1" key is perhaps the next most common choice for this function. This is a case in which propagating the use of a standard key, the "#" key, is highly useful. The caller must know what key to press, and it's awkward to prompt for this option in the caller interface. Yet, callers have no way of knowing what brand of voice mail system is answering their calls, or whether they've encountered voice mail or an answering machine! How is the caller to guess what key, if any, will provide them with editing functions?

Prompting may be the only reliable solution until the industry accepts a standard and use of the "#" key becomes widely known among users. On one vendor's systems, a guiding prompt to this effect is inserted after subscribers' call answering greetings based on a "mailbox class-of-service" setting. For example,

> Subscriber: *Hi this is Bill. I'm not in the office right now, please leave a message after the tone.*
> System: *After leaving your message, press "#" to edit your message.* *<Tone>*

Intelligent telephony agents and call-redirection systems pose an interesting problem. How disturbing is it to callers to dial someone they know, only to get an automated system trying to assert its own personality?

> System: *Hi, I'm the personal agent for*
> Subscriber (recorded): *Bill*.
> System: *If you'd like to leave a message, press 1.*
> *If you'd like me to page him, press 2,*
> *To call him at home, press 3.*

Until such systems become commonplace, we can expect them to be jarring to some users. A possible compromise is to allow subscribers the option to record the call-redirection options in their own voice, as a personalized greeting. Then, callers receive a greeting with the voice they expect, but also are offered the set of options that are available to them.

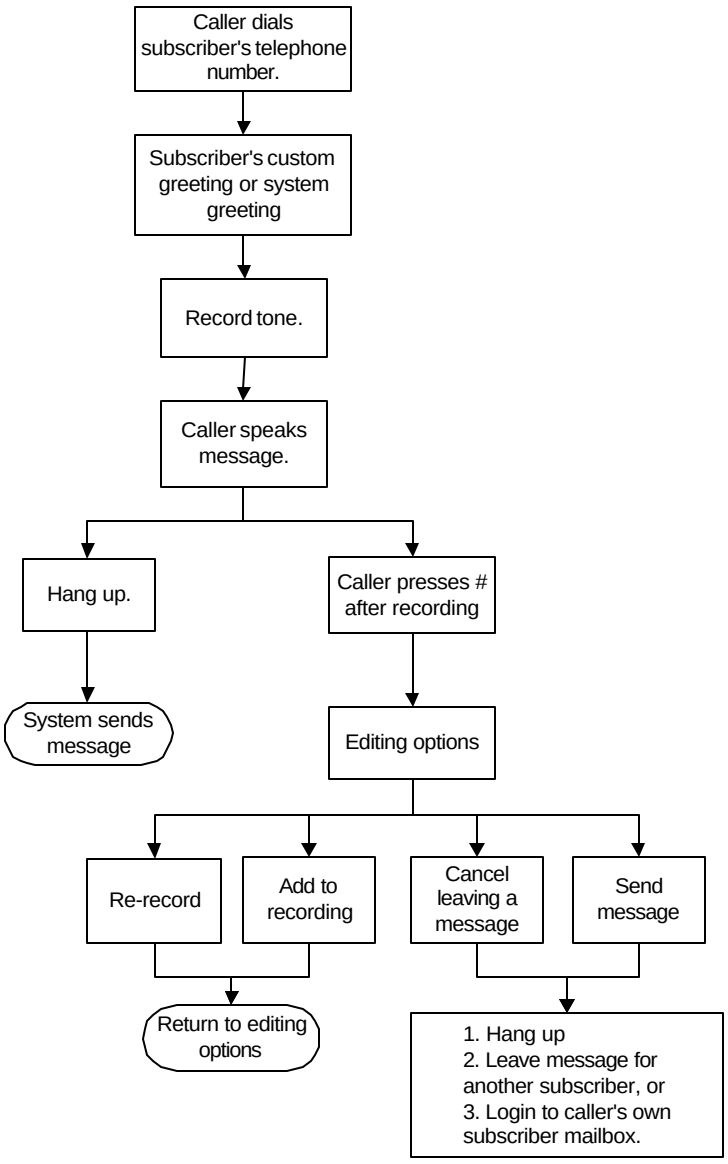

Figure 1. A typical flow of events in a caller interface. The transaction is like an answering machine unless the user enters a touch-tone (#)

2.2.1 Call Answering Greetings

In addition to editing functions, the voice mail system offers another value to the caller interface: customized subscriber greetings that change according to relevant states. The most common of these is to have a different greeting when the called party is busy versus not answering the phone.

> Subscriber greeting: *Hi, you've reached Bill. I'm on the other line right now. You can call back later, or leave a message after the tone.*

This is one way to give callers an option depending upon the urgency of their communication. Various voice mail systems offer additional greetings to be played after hours, on weekends, during extended vacations, or in the case of permanent absences. The other dimension is to allow the subscriber to pre-record alternate versions of each type of greeting. The complexity of this system can quickly multiply into what can be the most complex and confusing module of a voice mail system. As a result, customized greetings also risk being one of the least-used features of a voice mail system.

2.3 The Subscriber Interface

Retrieving, manipulating, and sending messages, and administering greetings and options are all part of the subscriber user interface, which is the user-system dialog experienced by the owner of the voice mailbox. Typically, all the functions are invoked from a *main menu,* which serves as a "home base" in the dialog. A typical menu structure is outlined in Figure 2.

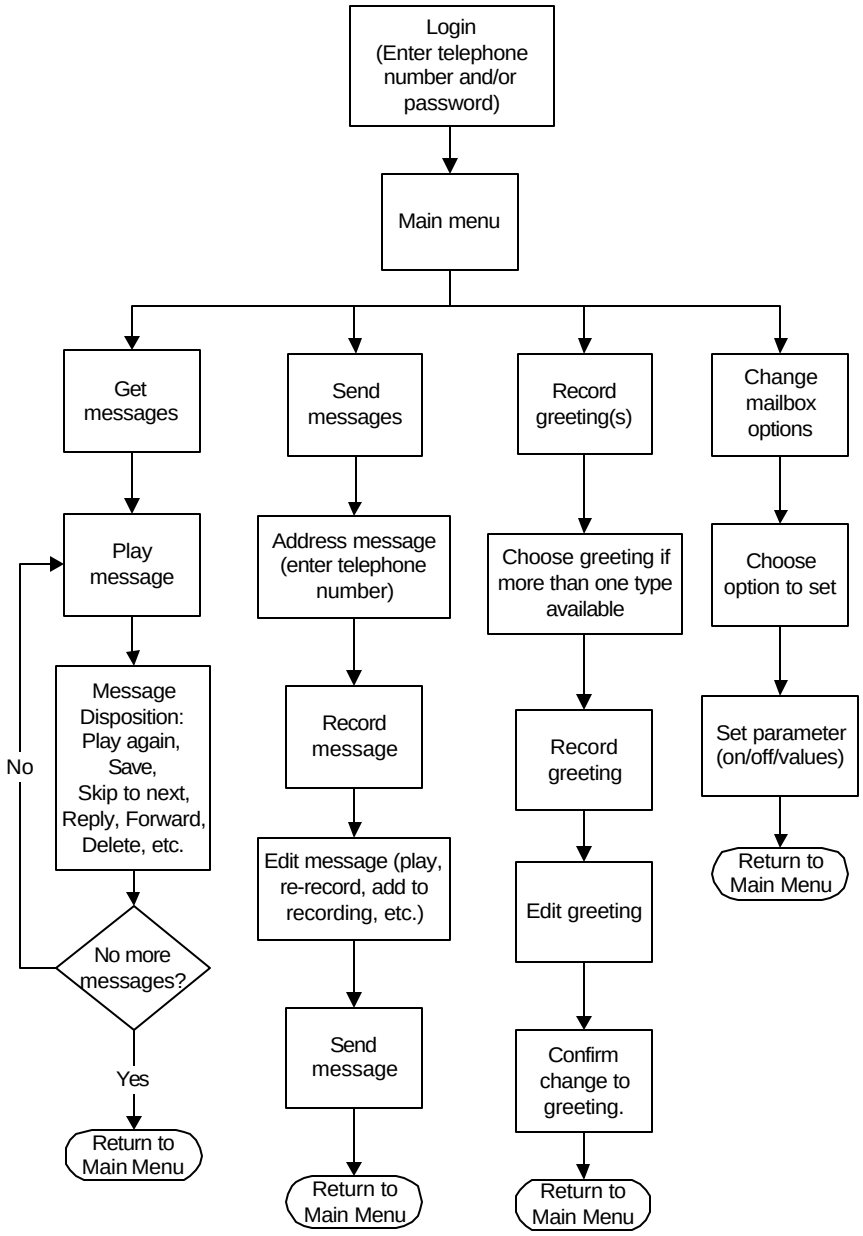

Figure 2. A very typical menu hierarchy for a voice mail subscriber user interface. The main menu is a "home base" to which the user returns after completing (or interrupting) a subtask

2.4 Retrieving and Manipulating Messages

Playing messages is the most basic and most frequent task for voice message system subscribers. Consequently, the steps needed to play a message should be minimized. Playing messages should be the first or prominent option in a main menu. After logging in, user should be required only to press one key, or no keys, to begin listening to their new messages.

Messages are categorized into at least two types: new messages and saved messages. Messages of these two types should be kept in separate *message queues*. That is, the user should be able to choose to listen to only new messages, only saved messages, or to both. New messages should be offered, or automatically played, first.

Another message category common to many systems is a "hold as new," "played but not saved," or "skipped" category. This class of messages should not be played in a separate queue. Rather, these messages are integrated with the new messages. Use of this category depends upon the default action, defined in the service, after the subscriber listens to a message. A system may automatically save a new message and move it to the saved queue if the user chooses to move to the next message. Alternatively, the system may keep the message as new if the user moves to the next message. The user must take explicit action to save the message, in order to move it to the save queue. A third design option is to save messages automatically, but provide the menu option of "keeping the message as new."

The analogy is to e-mail, where you can read new messages and keep them in your "inbox" before filing them or deleting them. This is a difficult concept for many users, and designers should consider dispensing with it if no significant user need is met by this "hold as new" class of messages.

A typical sequence for retrieving messages is as follows:

1. User selects "play messages" from main menu.
2. The first new message is played.
3. If there are no new messages, the first saved message is played.
4. During message playback, the user can speed up, slow down, skip ahead or replay the message also, the user can save, delete, or skip the message if they know the command.

5. After the message finishes playing, the user hears a menu giving options for disposition of the message that includes saving, deleting, replaying, replying to, and skipping the message.
6. Once the user indicates disposition of the message by a keypress, the system plays the next message.
7. If at the end of the new queue, the user is given the option of listening to saved messages.
8. At any time the user may exit the cycle of playing messages by hanging up or using a special keypress to go to the main menu or other system menu.

A default action on a message should always save the message, either saving as a new message or as a "saved" message in a saved queue. The default action should be performed if the message is interrupted by a hang-up or if the user chooses to go back to the main menu (or other menu) by pressing a key during message playback.

When playing new messages, the oldest new message should be played first, followed by other messages in order of their arrival time, until the newest message is played. One reason for this is to play sequential messages in the order in which they happened. For example, the caller may leave a first message that explains the background of a sales deal. The same user then leaves another message a hour later giving an update on the progress of the deal. When the user retrieves these messages, the newer message would be hard to understand without the background to the deal given in the earlier message. Chances are, the user might have to listen to both messages again to fully understand what was going on (thus doubling the time it takes to retrieve and understand the messages). If, however, the earliest message is played first, the background information is given first, and understanding follows simplistically.

2.5 Sending Messages

The promise of voice mail as more than a glorified answering machine in the network depends on the functions related to sending voice messages. Messages can be sent directly, with the subscriber entering one or more addresses and recording a voice message. Messages that already exist in a mailbox can also be sent using the reply and forward functions. If a message has been received from another subscriber, replying skips the addressing steps. Forwarding skips the recording steps, although many systems allow subscribers to attach a message of their own to forwarded messages.

A typical sequence for a subscriber sending a message is:

1. The user selects send or compose from the main menu.
2. The system begins the message-creating dialog.
3. The user is prompted for one or more addresses
4. The user is prompted to record the message to be sent.
5. Various post-recording options are offered.
6. The user gives input confirming that the message should be sent.

The order of addressing and message recording varies from system to system.

A little-discussed fact of voice messaging is that many users actually *don't send* messages. Unpublished AT&T survey data and actual usage data suggest that sending voice messages from the subscriber interface is, while not rare, a relatively less common voice messaging operation. (This has also been suggested from survey research by Rice & Tyler, 1995.)

Although sending voice messages seems to follow a fairly simple sequence of steps, there are actually a number of usability traps. Perhaps the most significant usability problem in sending involves positive transfer from the more familiar answering machine scenario. Many users simply hang-up after they are done recording a voice message. If the system send task order is record-then-address, this means that the message can't be sent, because no recipient has been defined. Even with the address-first order, most messaging systems don't send messages on hang-up. This problem doesn't go away with extensive experience with voice messaging on different systems, as both of us can attest.

Address entry also has some usability issues. Unlike call answering, where the user's greeting provides confirmation that the message will get to the right person, feedback for address in some voice messaging systems is simply the playback of the touch-tone digits the sender has entered, often without the benefit of appropriate pauses and intonation patterns that allow chunking of the numbers. On the system we use, positive acknowledgment of a successful address is marked by the absence of an error message following reading back of the number. Many PBX voice mail systems help with these problems by providing "nameback," playing the recipient's name as they recorded it as a confirmation.

Readers familiar with the GOMS family of task analysis (Card, Moran, and Newell, 1980, 1983; John and Kieras, 1994) might be able to see why it may actually make sense for voice mail subscribers to leave messages using call answering rather than sending messages. Using call answering, messages are "sent" by calling the person's number, waiting for the system to answer, listening to the greeting and the record tone, speaking the message, and hanging-up. Sending messages directly, from the voice mail subscriber interface, involves logging into the system (which can be skipped if the subscriber is already logged in), choosing send from the main menu, entering the address (often the same as the telephone number) and delimiting it, listening to nameback or numberback, listening the prompt for recording, recording the message and delimiting it, choosing among various send options (e.g., private and/or priority messages, future delivery), and, finally, confirming the send with a keypress. Informal data we collected from a variety of voice mail systems while preparing this paper while preparing this paper suggests that sending messages takes more time, involves more steps, and is more cognitively demanding than calling and leaving a call answering message.

One way around this is to encourage users to use the shortened task sequences of sending by replying and forwarding messages. Indeed, this is often a tip in corporate voice mail user guides.

2.6 Voice Messaging User Interface Standards

With the growth of voice messaging, people began to move from system to system and some to have multiple mailboxes. Because the various proprietary, usually PBX-based, business voice mail systems that emerged in the 1980s had different, often arbitrary, key bindings for core tasks, reports of destructive inconsistencies surfaced. What skips ahead within a message that you're listening to on one system deletes it on another. What lets a caller skip the greeting on most systems causes the system to say "fax included" for people who call me. After a decade of rapid growth in the 1980's, formal standardization of the touch-tone user interface to voice messaging became likely due to strong support for a standard among service provider, equipment manufacturer, and user group market segments.

Callers' user requirements were also a standardization driver. Callers can be thought of as *involuntary* users of systems who encounter someone's voice messaging system and leave messages. They often don't know and almost always don't care what kind of system they have reached. Many

callers use voice recording but no other features when leaving messages, but there are a half-dozen or so features that some number of callers do use. These include skipping greetings; listening to, re-recording, or deleting messages they are leaving; terminating their voice message; trying to reach a live person; getting help; forcing the system to disconnect; setting privacy or urgency options for a message, and logging in to their own mailbox after leaving a message. Industry and standards group participants believed that defining a common user interface for a few of these features could have large benefits to callers.

Voice messaging user interface standards work began in industry associations and moved to formal standards groups, ANSI in the US and ISO internationally. There was broad organizational participation in this work, with many issues of *what* should be included in a standard and *how* the standard should implement common features leading to contention and compromise. Once the international work began, this voice messaging user interface standard progressed quickly and was published in March of 1995 as ISO/IEC 13714 (ISO/IEC JTC1, 1995). Identical text with an ANSI cover and introductory material was published as ANSI/ISO 13714 in May of 1996. This standard is titled "User interface to telephone-based services: voice messaging applications."

The standard requires very little in terms of voice messaging features or their user interface:

- A record tone is required. One is recommended but no specific tone is mandated.
- Systems must offer context-sensitive help, which must be accessible after the user inputs a ∗0 or a 0 alone.
- The user must be able to dial-through system prompts, announcements, messages, etc. except for a few cases specifically called out in the standard.

A user interface standard can't really require *features* of a voice messaging system. Systems typically have record tones, context-sensitive help, and dial-through. More important, the standard defines required implementations for about a dozen core voice messaging activities.

Table 1. ISO/IEC 13714 standardized features

Feature	Required implementation
Caller interface	
1 Skip mailbox greeting	#
2 Delimit message recording	#
3 Delimit DTMF Input	#
4 Get system help	∗0
5 Force system disconnect	∗9
6 Has a record tone	Must be provided
7 Attendant/coverage	0
8 Dial-through	Must be provided
Subscriber interface	
(all above, plus)	
9 Return to main menu	∗7
10 Main menu: Listen	1
11 Main menu: Send	2
12 Listen menu: (Re)Play	1
13 Listen menu: Delete	3
14 Listen menu: Skip message	#
15 Send menu: Send message	4
16 Backup one 1 level or step	∗∗
17 Cancel voice or DTMF input	∗∗
18 Start voice recordings	#

The wording in standards differentiate required features from those which, if implemented, must have a specific user interface. In practice, since almost all of the features in this category are found on nearly all current systems, this is a strong requirement. The gist is that "if you provide capability *x*, it must by implemented as *y*, for your system to be conforming." These required implementations define a few basics of how users deal with messages, prompts, voice input, data, menus, and navigation. Depending on how the counting gets done, the standard covers about a dozen common voice messaging tasks. These are summarized in Table 1. ISO/IEC 13714 also collects and presents design guidelines and recommendations that reflect best industry practices for voice messaging user interface design.

In voice messaging, *callers* are people who encounter someone's system and leave messages. Although this can be as simple as dialing, waiting for answer and a record tone, then leaving a message, callers actually often need to be able to do more. It's useful to callers to be able to skip greetings. ISO/IEC 13714 promotes interoperability of systems for caller control of some basic functions. When systems conform, there will be fewer cases where what gets help on one system cancels message recording and causes a hang up on another, or where what skips through the greeting to the record tone on one system prompts the caller to login on another. Although in the big picture of voice messaging user/system interaction these caller features may not be used that often, the strong standards case for callers contributed to a relatively large number of caller features included. Several of the required implementations for callers came from defacto industry standards, at least in the U.S., on the use of # and 0.

For *subscribers*, most of the caller interface features standardized apply and the identical key bindings are required implementations. Most of what is additionally required for the subscriber interface are menu key bindings for the most common things people do when they access their voice mailboxes, particularly listening to and saving or deleting messages. If systems conformed, there would be fewer cases where what skips ahead on someone's work voice messaging system deletes it on their home system, or where what replays a message on one system deletes it on another.

With the publication of this formal user interface standard, voice messaging became perhaps the first application to have a comprehensive user interface standard. It is still too early to tell if this work will produce the user and industry benefits promised. There has been implementation of various parts of the standard by some vendors and some service providers, but legacy code, products, and even users have been highly resistant to change.

2.7 Alternative Approaches to Traditional Touch-tone Design

Voice mail systems as well as most other IVR systems use a fairly standard and ubiquitous dialog technique: a series of prompts with options, i.e., a "menu" format. We have assumed that the voice mail application follows this dialog structure in our discussion so far. But what about some different, innovative approaches to touchtone user-system interaction? For example, in the IBM Speech Filing System (an early IVR application

incorporating voice mail and voice reminder functionality), Gould and Boies (1983) designed what was basically a touchtone command system. The user entered two-key touchtone sequences without being prompted for those sequences. There were no prompts but there was feedback given as to which command was invoked. So, for example, to compose and send a message, the user would press ∗R to record, the system responds with "Record", the user records their message, then presses ∗T to send (transmit) the message, the system responds by saying "Transmit", then the user enters the telephone number of the recipient, and so on. Later, Gould and Boies (1983) switched to the modern prompted menu format. Although they noted that the menu format allowed users to use their system without instruction, the speed of interaction suffered. Over the last decade, the advantage of the menu approach, in allowing voice mail to be used without detailed instruction, has overwhelmed any advantage in speed obtainable from a command approach. Using type-ahead, expert users can also simulate a command approach. However, a menu structure optimized for menu use is not optimally efficient for a command approach. For systems that have a large percentage of expert users, like voicemail, it might be worthwhile to revisit the command approach. The best technique might be to incorporate elements of commands within a menu system, to allow both novice and expert use. For example, a set of ∗-digit command sequences could be active throughout the interface to invoke a set of commonly used functions.

Recently, Resnick and Virzi (1995) proposed a number of other dialog formats for touch-tone systems as possible alternatives to the common menu design. They proposed a variety of different techniques, some of which would be more applicable to IVR systems that have simple transactions and do not have expert users. None of their schemes were extensively tested with users, but certainly different approaches may be worthwhile exploring as novel solutions to certain interactions. For example, Resnick and Virzi proposed one scheme they call a "rejection menu dialog," which is kind of a twenty-questions approach to menu choice. Using voice mail functions, one rejection menu dialog might be as follows:

System: *If you'd like to get messages, please wait, otherwise, press #.*
User: #
System: *Send a message. Wait to select or press # to reject.*
User: #
System: *Change your greeting. Wait to select or press # to reject.*
[etc.]

Obviously, this scheme does not transfer well to the main dialog in a voicemail system. However, it may be more appropriate for simple two- or three-choice options, which do occur at many points in voicemail systems. (The rejection dialog might also be cast as a sequence of yes/no or two-choice option questions.)

Overall, alternative dialog techniques such as commands or rejection menus may be useful to explore in voicemail systems as *supplements* to the main design rather than substitutes for the standard menu approach.

3. AUTOMATIC SPEECH RECOGNITION AND VOICE MAIL

The next technology for interaction with computers over the phone is speech recognition by machine. Voice commands can supplement or replace the use of the touch-tone keypad in controlling a voice mail application. Elsewhere in this book (Chapters 7 and 13) the problems and challenges associated with using speech recognition in IVR have been discussed, and certainly these same considerations apply to voice mail applications.

In addition to usability considerations for automatic speech recognition systems in general, voice mail presents two problems that are not often significant in other telephone applications:

1. Voice mail systems are functionally complex and thus require complex vocabularies.

2. The system must distinguish between times when the user is talking to the system and when the user is recording a message.

Functional complexity places a burden both on the user and on the speech recognizer. The user must remember a long list of commands, which means either mastering the keywords of a system, or paying close attention to menus that exhaustively list commands and their functions. For the speech recognition system, large vocabularies may reduce the effectiveness and accuracy of the system. Depending upon the technology, the system may perform better with phrases than single word commands (e.g., “Help me out” rather than just “help”), but this imposes an even further onerous memory burden on the user. The solution to this is more flexible natural language systems (cf., Chapter 2) which accept many variations on natural phrases.

A fruitful technique for dealing with experts versus novice users of a voice command system is used by Brems, Rabin, & Waggett (1995). In essence, Brems et al. suggest:

1. Choose commands that are transparently descriptive of their function.
2. Choose a technology that allows barge-in of voice commands.
3. Prompt only for the commands first.
4. Pause for experts to respond. This allows expert users who know the commands to respond right away without having to listen to a long explanation.
5. Follow the pause with a more detailed menu for novice users.

For example, applying the Brems et al. suggestions to voice mail, a menu might be presented this way:

> [System plays message]
> System: *Please say delete, reply, call back, forward, play again, or next message.*
> <pause>
> System: *To erase this message, say delete, to answer this message by voice mail, say reply, …*

The ability to skip over long prompt explanations is a crucial element for giving a speech recognition user interface an advantage over the touch-tone alternative. Franzke, Marx, Roberts, and Engelbeck (1993) found that users of a speech recognition user interface to a voice mail system completed tasks faster than users of a touch-tone version of the same system. The main reason for the advantage was that speech users learned a set of simple commands and readily interrupted the prompts, whereas touch-tone users tended to listen to prompts. (Note, though, these were novice users observed only through a short experiment, and that Fay (1993) and Karis (1997) found the opposite preference with different applications.)

Also a challenge both to usability and technology is the need to distinguish when the user is addressing a command to the system or recording a message. To illustrate, imagine a system accepts the command "erase" while the user is making a recording. If the word erase appears in the message content (e.g., "we'd like to *erase* our third quarter losses."), the message is erroneously deleted.

This restricts the user to the use of a single keyword that catches the attention of the recognizer. (Note that this is a loss of functionality relative to a touch-tone user interface, which can allow touch-tone recording control during message recording.) The speech recognizer will monitor for one keyword or phrase such as “Ready”, “End recording”, “Wake up”, or “Computer!”. There are two technical problems with this, one is the burden on the processor, which must continuously recognize the speech stream in order to monitor for the wake-up command, and the other is finding a word which makes sense to users but which is unlikely to be used in conversation. (Compare, for example, the Wildfire call management system, which monitors for the word “wildfire” during normal telephone conversations, presumably with the hope that wildfires are not a common subject of business discourse.)

Of course, the drawback here is that the user must remember to use the wake-up keyword without immediate prompting. Depending upon the capabilities of the technology used, it may be more practical to use a touch-tone command to terminate a recording. But if the ASR system is being targeted to users of rotary telephones, this poses a problem. Recordings can terminate via a time-out period, but this could quickly become tedious to users, because the time-out interval must be longer than pauses found in normal conversation.

Fay (1993) and Karis (1997) have found that users may actually prefer touch-tone based systems to voice command systems, due at least in part to the limitation of speech recognition technology. From early applications of the technology, human factors experts have noted (e.g. McCauley, 1984) that users may not tolerate recognition errors associated with ASR unless there is a distinct advantage to be obtained from using voice control. Most often cited is the ability to control a device in a “hands-free” mode. In the domain of voice mail, the need for hands-free control is well-noted for accessing voice mail from a cellular car phone. This is a natural usability advantage that can be exploited to increase the acceptance of ASR technology.

4. UNIFIED MESSAGING AND MULTIMEDIA MAIL

Ironically, voice mail began as a sort of telephone analog of the terminal-based electronic mail systems of a decade ago. Now, the similarities between voice mail and e-mail are pushing the technologies to merge into *unified*

messaging. Unified messaging, in its ultimate form, is the combination of voice and electronic messages and faxes into one single mailbox, which can be accessed through an IVR system over the telephone or through a mail client on a computer. A review of recent unified messaging platforms and service can be found in Kosiur (1998).

4.1 Fax Messaging

This trend certainly began when voice mail systems began to add a premium feature: the ability to intercept and electronically store faxes with the user's voice messages. Faxes can be received by the voice mail system in two ways:

- Faxes are sent to the user's regular telephone number. When the voice mail system answers the call, it detects the fax tone, receives the fax electronically, and stores the fax as a special class of message.

- Faxes are sent to a second number, a fax number. The voice mail system again stores incoming faxes electronically. The fax messages then are transferred to the user's voice mailbox.

Alternative configurations of these fax features are also offered by vendors to solve various office problems. So, for example, a busy fax machine can be given a voice mailbox. When a fax comes in while the machine is already receiving the fax, the fax machine's voice mail picks up and stores the new incoming fax for later printing.

Fax call answering presents an enormous usability problem that the industry has yet to solve effectively. When the system uses the regular voice telephone number to receive fax messages, things work well as long as the user doesn't answer their phone. However, it is not possible to detect a fax call before an incoming call is picked up. Thus, subscribers inevitably must tolerate getting a fax tone sometimes when they answer a call.

A second solution, using a separate fax number, thus may be better from a usability point of view. (The user's callers must know two numbers to call for fax versus voice, but that is a very familiar convention.) Unfortunately, this solution is expensive, for there must be either a separate extension for every person in the organization (thus doubling the needed capacity of the phone system) or the fax mailbox must be shared among users. The shared fax mailbox is also expensive in terms of human time, for a person must

manually view the faxes and then forward them to the appropriate user mailboxes. This remains a challenge for fax and unified messaging.

4.2 Viewing Voice Mail

The next voice mail feature, which has been offered as a separate component, is the ability to view a list of voice mail messages on a personal computer and to listen to them (provided the user's computer has sound capability). At least part of the rationale for offering the ability to view voice mail headers is based upon a human limitation: it is difficult to manipulate lists of items using the auditory medium. Using the telephone, the user is forced to search messages one-by-one in a fixed sequence. The user may also need to rely on memory to recall the relative order of messages in the list. The user might be able to skip over messages but still must go through them in serial order. Systems can be designed to let users to jump to a message using a "message number", e.g.,

System: *You have twenty saved messages.*
What message do you want to hear?
<pause> enter the number of the message.
User enters: 17
System: *Message 17. July 7, 1998. …*

Obviously, this poses an impractical memory burden on the user. Can we expect users to know the order in which they saved their messages? Another option is to provide audio skimming and search capability through speech processing technologies (Marx & Schmandt, 1996; Roy & Schmandt, 1996). These are promising technique that could be applied to voice mail retrieval, but they are still in the research stage.

This difficulty is eliminated if the user is able to see a visual list of their messages. The user can scan all the messages in a visual list and does not need to recall the order of a message in a list. It is then a simple matter to choose the message to manage (play, delete, reply to, forward, or file the message) from a complete list. The message could be manipulated from the computer user interface or from the telephone IVR system. Fusco & Gattuso (1991) found that no only did users prefer to use a visual interface when using the more complex features of voice mail, but also users found the voice user interface less confusing after using the visual user interface.

This same capability has also been offered on display phones, where a list of voice message headers can appear on a large display on the telephone. Buttons next to the display, called “soft keys”, allow users to choose a message to manage. Such a system is most usable if there is a tight and elegant integration between the telephony user interface and the display.

For example, the user is presented with a list of messages, one per line, with a soft key to the right of each message header. Soft keys at the bottom of the display are labeled “delete” “reply” and so forth. While the user holds the handset and presses a message button, the message is played. After playing, a touch-tone menu is played. The user can press the “delete” soft-key over the menu in order to quickly delete the message just played. The response should be immediate and real time with the IVR system.

Soft key systems have not proliferated, however, particularly in the face of personal computers. Because, particularly in the business environment, a user is likely to have a personal computer, and that computer is likely to be close to the telephone, computer-based applications are more practical.

Having a separate application for voice mail may seem redundant, since the major functions are identical to an e-mail client program. Hence, unified message seems likely to supplant separate applications that display voice mail messages.

4.3 Listening to E-mail

It has long been possible to listen to electronic mail over the telephone using text-to-speech (TTS) technology, and, indeed, AT&T Mail, among others, had offered MailTALK℠ for its customers eight years ago. Perhaps because of the greater use of e-mail over the last few years (connected with the prominence of the Internet), there has been a resurgence of interest in reading e-mail over the telephone.

There are four notable usability considerations with e-mail over the telephone:

1. *The acceptability of text-to-speech technology.* Even with the technology of a few years ago, text-to-speech is highly intelligible. Users are nearly as accurate as with natural speech in understanding TTS (Paris, Gilson, Thomas, & Silver, 1995; Pisoni, Manous, & Dedina, 1987; Syrdal, 1994). Listening to TTS may require more mental processing, and can

be more exhausting than natural speech to listen to in large blocks. The real user impediment to TTS, however, is that users simply dislike the sound of TTS regardless of how intelligible it is (Cowley & Jones, 1992).

2. *Transliteration of textual conventions to voice*. E-mail messages are full of textual conventions that are visual in nature, and do not translate to speech directly. Smiley faces (e.g., ;-) are an obvious example, but even more ubiquitous are lines of dashes or stars used as rules or demarkations. The system must be suppressed in some manner to prevent it saying something like the word "dash" eighty times in the middle of a message. E-mail messages also have headers with detailed routing information. Reading of those headers should also be suppressed. The common way this suppression is accomplished is by pre-processing the e-mail text before it is given to the TTS processor. A significant improvement to usability can be obtained with more sophisticated pre-processing.

3. *The sheer volume of e-mail messages for some users*. For many heavy, or even moderate, users, the number and frequency of e-mail messages far outstrips the number of voice mail messages one might typically expect to receive. Where it might make sense to light a message light to signal the arrival of voice mail, chances are it makes no sense to light a message light on a telephone when the user receives e-mail, since that light would be on constantly. The high volume of e-mail poses problems for inventory of messages, user navigation among messages, filtering through relevant as opposed to junk messages, and time spent listening to messages.

4. *Navigation among e-mail messages*. Earlier we mentioned the difficulty of randomly accessing voice mail messages when using a telephone-based user interface. Messages must be accessed sequentially. The same problem is just as true when moving among e-mail messages over the telephone.

An application that reads e-mail is very much like a voice mail application in its structure and functionality. The next logical step is to combine e-mail and voice mail into the same application.

4.4 Putting it All Together

Unified messaging puts all the above features together, with the merging of all message types into one common mailbox. Voice, e-mail, and fax messages are stored in the same mailbox. A computer program, most likely an e-mail client, can access the mailbox. Voice, e-mail, and fax messages are all shown in the message list, and voice messages can be played by the computer (provided proper hardware and software is present, of course). But in addition, there is a telephone application, the extension of a voice mail application, which plays voice messages, reads e-mail messages by TTS, and lists fax messages and forwards them for printing.

In addition to all the usability considerations for the separate elements discussed above, unified messaging poses the additional usability question: how much access to messages do users really want?

This is particularly relevant to telephone access to the unified message mailbox. Access to three kinds of messages means triple the message queues to listen through. Separating message types must be done, as e-mail will overwhelm voice and fax in volume. Some users may dislike e-mail TTS. Extra menus must be added to a voice mail application framework in order to navigate among the message types.

Perhaps as users adapt to the functions offered by unified messaging, it is best to provide a wealth of user customization of the telephone application. For example, allow users to suppress the access to any of the three message types. Some users could configure their unified voice mailbox to just get their voice mail, making it like their more familiar voice mail. Others could get only voice and fax messages, other users could choose to get all three types. Allow configuration of e-mail access, so that a user who dislikes TTS can skip over the reading of e-mail message bodies, or have an option that only reads e-mail headers.

4.5 Mixed Media

The unified mailbox is really a multimedia mailbox: visual and auditory messages are mixed in the user interface. The user can use a visual (computer) user interface or an audio (telephone) user interface to work with those voice and visual messages. A simple extension of such a design is to allow mixing of media in a single message. Thus, the user can receive a fax that is prefaced by a voice annotation.

The next opportunity that presents itself is the freedom to respond to messages in any mode. Thus, the user receives a voice mail message through their computer user interface, and decides to respond with an e-mail. The recipient of the response, if at their phone, can still listen to the message, since the e-mail can be read by text-to-speech.

How valuable is this? This is the major challenge to user interface designers for the next generation of messaging: how do people want to get their messages, and do they need the freedom of mixed media at the cost of increasing complexity of the user interface?

It has often been observed informally that there are individual differences in preference for visual versus auditory communication. Some users prefer to use e-mail, others prefer to use voice mail, and one learns in an organization the optimal method to contact certain individuals. The phenomenon is ripe for some systematic study. Multimedia messaging allows users to communicate in their own preferred media rather than that of their correspondent. But is this really a problem? Will users simply tend to reply using the mode by which they were first contacted?

Of course, the other variety of multimedia mail is call answering for videotelephones. If videotelephony ever becomes common, then videotelephony mail will follow. The video mail system will have new user interface challenges based upon the capabilities of the videotelephone device (be it a telephone and/or a computer). Video mail will also then become another medium to manage in a unified mailbox. However the existence of video mail as a consumer service is dependent upon the regular use of person-to-person video calling (as opposed to arranged video conferences), and the history of attempts to introduce person-to-person video calls is not encouraging (Blanchard & Angiolillo, 1994; Noll, 1992).

REFERENCES

Blanchard, H. E., & Angiolillo, J. S. (1994). Visual displays in communications: A review of effects on human performance and preference. In *SID International Symposium Digest of Technical Papers, 25*, 375-378.

Brems, D. J., Rabin, M. D., & Waggett, J. L. (1995). Using natural language conventions in the user interface design of automatic speech recognition systems. *Human Factors*, *37*(2), 265-282.

Card, S. K., Moran, T. P, & Newell, A. (1980). The keystroke-level model for user performance with interactive systems. *Communications of the ACM, 23*, 396-410.

Card, S. K., Moran, T. P., and Newell, A. (1983). *The Psychology of Human Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Clark, H. H. and Haviland, S. E. (1977). Comprehension and the given-new contract. In R. O. Freedle (Ed.), *Discourse production and comprehension* (pp. 1-40). Norwood, N.J.: Ablex.

Cowley, C. K., & Jones, D. M. (1992). Synthesized or digitized? A guide to the use of computer speech. *Applied Ergonomics*, *23*(3), 172-176.

Englebeck, G., & Roberts, T. (1990) The effects of several voice-menu characteristics on menu-selection performance. In *Proceedings of the Bellcore Symposium on User Centered Design* (pp. 50-63). Bellcore Special Report SR-STS-001658. Red Bank, NJ: Bell Communications Research.

Fay, D. (1993). Interfaces to automated telephone services: Do users prefer touchtone or automatic speech recognition? In *Proceedings of the 14th International Symposium on Human Factors in Telecommunications* (pp. 339-349). Darmstadt, Germany: R. v. Decker's Verlag.

Franzke, M., Marx, A. N., Roberts, R. L, & Engelbeck, G. E. (1993). Is speech recognition usable? An exploration of the usability of a speech-based voice mail interface. *SIGCHI Bulletin*, *25*(3), 49-51.

Fusco, M. & Gattuso, N. (1991, September). *An assessment of a visual voice mail interface*. Poster presented at the 35th Annual Meeting of the Human Factors Society, San Francisco, CA.

Gould, J. D, & Boies, S. J. (1983). Human factors challenges in creating a principal support office system—The speech filing system approach. *ACM Transactions on Office Information Systems*, *1*(4), 273-298.

Halstead-Nussloch, R. (1989). The design of phone-based interfaces for consumers. In *Proceedings of CHI-89: Human Factors in Computing Systems* (pp. 347-352). New York: ACM.

John, B. E., & Kieras, D. E. (1994). *The GOMS family of analysis techniques: Tools for design and evaluation.* Carnegie Mellon University Human Computer Interaction Institute Technical Report 94-106.

ISO/IEC JTC1 (1995). *ISO/IEC 13714: User interface to telephone-based services: Voice messaging applications.*

Karis, D. (1997). Speech recognition systems: Performance, preference, and design. In *Proceedings of the 16th International Symposium on Human Factors in Telecommunications*. Oslo, Norway.

Kosiur, D. (1998). Making life simpler with a universal mailbox. *Internet Computing*, *3*(9), 87-89.

Lee, E., & MacGregor, J. (1985). Minimizing users' search time in menu-retrieval systems. *Human Factors*, *27*(2), 157-162.

Leopold, F. F., & van Nes, F. L. (1984). Control of voice mail systems by voice commands. *IPO Annual Progress Report, 19*, 118-122.

Marics, M. A., & Engelbeck, G. (1998). Designing voice menu applications for telephones. In M.G. Helander, T.K. Landauer, & P. Prabhu (Eds.), *Handbook of Human-Computer Interaction*, Second Edition. New York: Elsevier.

Marx, M., & Schmandt, C. (1996). MailCall: Message presentation and navigation in a non-visual environment. In *Proceedings of CHI-96: Human Factors in Computing Systems* (pp. 165-172). New York: ACM.

McCauley, M. E. (1994). Human factors in voice technology. In F. A. Muckler (Ed.), *Human factors review: 1984* (pp. 131-166). Santa Monica, CA: Human Factors Society.

Noll, A. M. (1992). Anatomy of a failure: Picturephone revisited. *Telecommunications Policy*, *18*, 307-316.

Norman, K. L. (1991). *The psychology of menu selection: Designing cognitive control at the human/computer interface*. Norwood, NJ: Ablex.

Paris, C. R., Gilson, R. D., Thomas, M. H., & Silver, N . C. (1995). Effect of synthetic voice intelligibility on speech comprehension. *Human Factors*, *37*(2), 335-340.

Pisoni, D. B., Manous, L. M., & Dedina, J. J. (1987). Comprehension of natural and synthetic speech: Effects of predictability on the verification of sentences controlled for intelligibility. *Computer Speech and Language, 2*, 303-320.

Resnick, P., & Virzi, R. A. (1992). Skip and scan: Cleaning up telephone interfaces. In *Proceedings of CHI-92: Human Factors in Computing Systems* (pp. 419-426). New York: ACM.

Resnick, P., & Virzi, R. A. (1995). Relief from the audio interface blues: Expanding the spectrum of menu, list, and from styles. *ACM Transactions on Computer-Human Interaction*, *2*(2), 145-176.

Rice, R. E., & Tyler, J. (1995). Individual and organizational influences on voice mail use and evaluation. *Behaviour & Information Technology, 14*(6), 329-141.

Roy, D. K., & Schmandt, C. (1996). NewsComm: A hand-held interface for interactive access to to structured audio. In *Proceedings of CHI-96: Human Factors in Computing Systems* (pp. 173-180). New York: ACM.

Schumacher, Jr. R. M., Hardzinski, M. L., & Schwartz, A. L. (1995). Increasing the usability of interactive voice response systems: Research and guidelines for phone-based interfaces. *Human Factors, 37*(2), 251-264.

Syrdal, A. K. (1994) Development of a female voice for a concatenative text-to-speech synthesis system. *Current Topics in Acoustic Research, 1*, 169-181.