Phoneshell: the Telephone as Computer Terminal

Chris Schmandt
M.I.T. Media Laboratory

ABSTRACT

This paper describes Phoneshell, a telephone based application providing remote voice access to personal desktop databases such as voice mail, email, calendar, and rolodex. Several forms of information can also be faxed on demand. Phoneshell offers its users numerous opportunities to record voice entries into its underlying databases; this new utility for stored voice as a data type, necessitates multimedia support for the traditional graphical user interfaces to these same databases. The experiences of a small Phoneshell user community are discussed, with emphasis on key features which are most important to its success. The underlying software architecture used by Phoneshell includes a toolkit for building interactive telephone-based services.

Keywords: digitized speech, speech synthesis, interactive voice response, auditory user interface.

A SCENARIO

At the airport on the way home your flight is posted with a 20 minute delay, hardly time to hook up a modem; instead, you just call your office computer on the nearest pay phone. After hearing a new voice mail message, you turn your attention to your email; it has been filtered and you have time for only the most important messages, which are read to you by a speech synthesizer. Darn, that new sponsor wants to come visit next Tuesday. You check your calendar, and see that you are free except for lunch with a friend from across town. You record an entry for the sponsor visit into the calendar so that your secretary will keep the day free. Then you switch to your rolodex, and look up your friend. Since you don't have time for a conversation, you don't call her to reschedule lunch, but instead record a voice message which will be sent to her as email. You have no additional important messages, but without hanging up the phone you call home through your own rolodex, and leave a message on the machine there saying that you'll miss dinner. Elapsed time: 4 minutes; still time to find some inedible airport food.

INTRODUCTION

Most workstations and personal computers now include speakers and microphones, supporting voice digitization and playback without additional hardware. Microprocessors are fast enough to support limited vocabulary speech recognition, full text-to-speech synthesis, and a variety of voice processing techniques entirely in software. Workstations are beginning to appear with built-in ISDN digital telephone network interfaces. Audio, both as a data type as well as a component of the user interface, can easily be supported within the the bandwidth and storage capabilities of current networks and magnetic media. Voice should be appearing on every desktop, and some applications, such as voice control of windows (e.g., [12]), can already be found across many platforms.

Speech recognition enhances a user's performance in situations where his hands and eyes are busy or when he can perform multiple tasks using different modalities [5]. Voice is a powerful communication medium, effective for problem solving [7] and rich in expressiveness [2]. We speak much faster than we can type. But in fact speech applications are lagging the ability of technology to support them. For example, although several vendors support voice attachments to normal electronic mail, and a vendor-independent standard for such messages, MIME [10], seems to be emerging, communities of heavy email users generally have not taken advantage of these voice attachments.

Stored voice is very difficult to utilize effectively. Listening to speech is much slower than reading for most of us, and when reading we can more easily skim and let our eyes wander while speech must be accessed serially. Speech is transient and requires our attention; while a screenful of text remains available until we have a chance to look at it, audio output is lost unless we attend to it at the moment it is spoken. Voice messages are hard to file as we cannot yet index them using keyword searching.

Voice is much more likely to be successful in situations which exploit its particular advantages, especially our ability to use voice without a screen or keyboard; such situations include telephone-based remote access to the office as well as emerging hand-held computing appliances [15]. But just as we are accustomed to using many desktop applications simultaneously through window systems, such remote or portable access will be more powerful if it, too, can allow interaction with a full suite of applications. And as we begin to utilize voice in these new computing environments, it will impact applications on the traditional office workstation as they, too, will be increasingly required to manage stored voice as a data type.

This paper describes Phoneshell, a telephone-based speech interface to personal information management functions. Using digitized speech as well as text-to-speech synthesis and touch tone input, Phoneshell provides interactive access to voice and electronic mail, a personal calendar, individual and lab-wide name and address databases with speed dialing, facsimile services, and public information sources such as news, traffic, and weather. To send email or add an entry to the calendar a user records a voice file which is either embedded in an email message or associated with a date and time in the calendar database. As a consequence, these screen-based email and calendar applications must now support multiple media in their visual user interfaces; users need to retrieve recorded voice at the desktop as well as over the telephone.

This paper first describes Phoneshell, in terms of both its functionality all well as its user interface. It then describes the new versions of desktop applications designed to handle the multimedia databases resulting from telephone integration of voice and text. Although not a formal field study, a description of user experiences with Phoneshell over several years indicates that systems such as this can successfully fill real communications needs. Finally, the underlying architecture and supporting audio interaction toolkit used by Phoneshell is described.

Integration is the dominant theme of this paper. Taken in isolation there is little that is novel in the work described here; applications similar to pieces of the Phoneshell environment have previously been built at M.I.T. and elsewhere. What is novel is the unified environment making voice available across many applications which are also accessible over the telephone. The telephone interface is enhanced by supporting a full range of desktop applications so that one can, for example, hear a text message about a meeting, consult and modify one's calendar, and send a voice message or place an onward call as an acknowledgment during a single phone call. Screen-based applications are made more powerful when voice becomes a flexible data type with a consistent user interface in which a user is given interactive control over sound playback, visual cues to sound contents, and the ability to move sound snippets between applications.

PHONESHELL

A user calls in to Phoneshell using an ordinary touch tone telephone and during the call invokes a number of constituent applications. Each application uses touch tones for input, and a combination of speech synthesis, digital recording, and playback for responses. In many ways Phoneshell is like the increasingly common interactive voice response (IVR) systems which provide access to information such as flight schedules, film showing times, and bank balances, although the information Phoneshell presents is particularly timely and personal. Phoneshell is different from these IVR applications in that it can take action for the caller, including recording messages into databases, placing onward telephone calls, and sending files as facsimile. The layout of Phoneshell's menu options at the top level is shown in figure 1.

Phoneshell applications are developed independently but must finally be compiled into a single executable file; a table contains entries describing each application so that a top level menu can be automatically generated. Each application supports a simple common programmatic interface, with entries for initialization (when a new user calls in), execution (when the user selects this application, possibly multiple times during a call) and termination (on completion of a call). Currently six applications have been developed for Phoneshell: voice mail, email access, calendar, rolodex, lab-wide dial-by-name, personal name and address database, and an information application for weather, news, and traffic access. A seventh application under development can report who is present in the lab and call such users at the nearest telephone.

Voice mail

The Phoneshell voice mail application is similar to many commercial voice mail systems; voice mail is more interesting in terms of its graphical user interface, described below. Office phones forward to an ISDN based voice mail server which plays greetings and records messages; when a new message is taken, email notification is sent to the recipient [14].[1] Incoming voice messages embedded in Internet mail are also routed to a user's voice mailbox. A Phoneshell user can hear new or old messages, record messages to other voice mail subscribers, or change the voice mail greeting. Voice messages can be played back at a faster rate than they were

[1] Besides alerting users who are in other offices or at home, this email can be used to trigger messages to alphanumeric pagers.

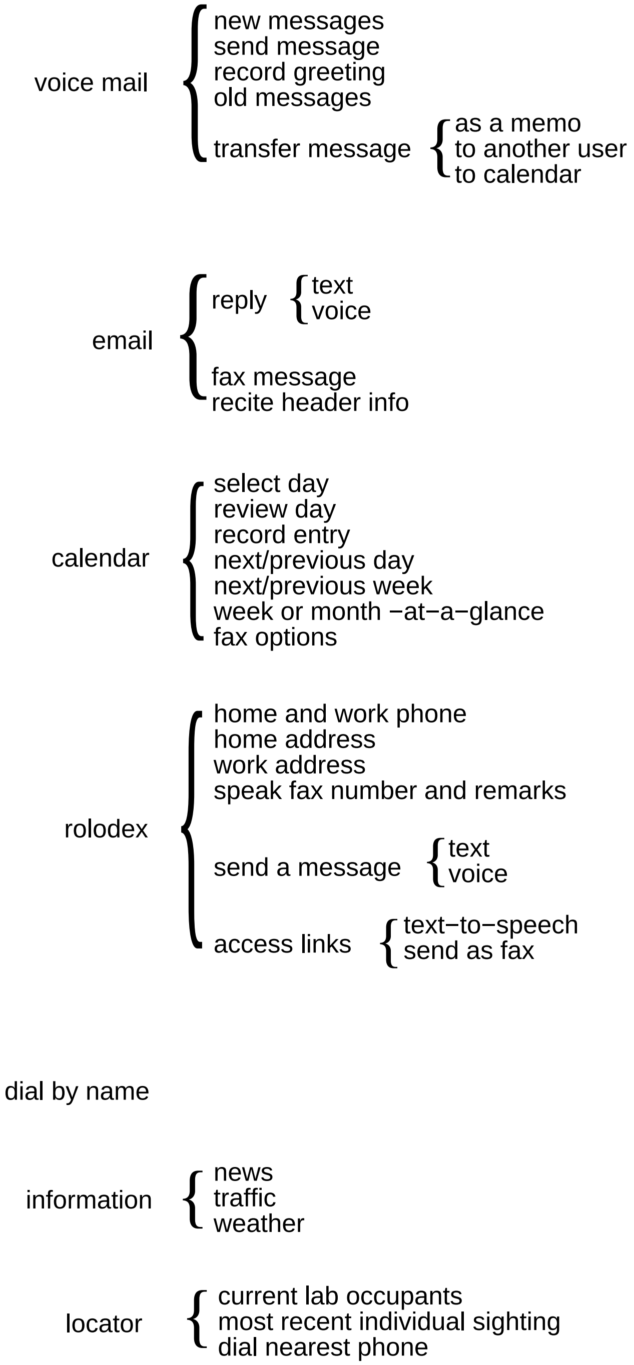


Figure 1: Phoneshell's menus provide access to a variety of personal information management functions.

```
To: Chris Schmandt <geek@media.mit.edu>
Subject: Re: techno-Bill
In-Reply-To: Your message of "4 Jan 1993 13:06:57 EST"
             <9301041806.AA07990@media.mit.edu>
X-Mailer: XVMH [1.0.4, MH 6.7]
Date: Mon, 04 Jan 1993 14:09:18 -0800
From: "Ben Stoltz" <Ben.Stoltz@Eng.Sun.COM>
Content-Length: 371
Status: RO


>Chris Schmandt <geek@media.mit.edu> writes:
>I sent you a fax of a comic strip, but it probably
>didn't >come out well.  Let me know -- now that I'm
>at work and can access real copying machines I can
>probably do better!
>chris

The last frame did not come out very well. The
punch-line is missing!!
I do get the general idea though:^)
Please send it again.
```

Figure 2: Email messages may include lengthy headers and embedded text.

spoken without pitch distortion (the algorithm, SOLA, is described in [11]). A per-user profile specifies the rate, with 1.4 times faster being common for frequent users; the specified playback rate is used for whenever any Phoneshell application plays a voice file. Users can also record voice "memos", which are automatically incorporated into a screen-based reminder list application. While listening to voice messages, they can be transferred to another user, converted to a memo, or copied into the user's calendar (in which case a date is entered with touch tones).

Text mail

The email application sorts mail, presents it with text-to-speech synthesis, and can generate voice or text replies. Filtering categories are specified by the user in a profile, which allows sorting based on regular expression matching against the sender and subject lines as well as the message body.[2] In many mail filtering systems, such as the seminal Information Lens [4], the filter routes mail to particular folders which the user reads in any order. For the mail reader, filtering is invoked at the time the user phones in, and determines the presentation order of messages.[3] For example the author uses categories of "very important" (sponsors, lab director), "important" (lab research group members, certain lab administrators), "personal" (addressed to the recipient by name instead of a mailing list) and "other", and a small number of messages are automatically deleted.

[2] Filtering is based on a modified version of *procmail*, a public domain mail filter.

[3] The same filter software is also executed, with different rules, whenever a new mail message arrives. This version of the filter forwards some messages to an alphanumeric pager, and also separates out multimedia messages with voice annotation, so they can be accessed with the same user interfaces as provided for voice messages from the telephone.

Once sorted, messages are summarized. For each message the user hears the name of sender, the subject, and an indication of the length; for example “Long message from ‘Don Jackson’ about ‘visit next week?’ ” If the user chooses to hear the message, it is broken into sentences for synthesis, and touch tone keys can repeat a sentence or skip to the next one. The sentence parser also detects blocks of text included from prior messages (figure 2); during presentation these area announced (“included text follows...”) and may be skipped over. Included email headers are further analyzed and summarized instead of being spoken in excrutiating detail. As with voice message playback, a per-user profile selects the speech rate for synthesis of all Phoneshell text output; although the default rate of the speech synthesizer is 180 words per minute, frequent users generally set the rate to between 280 and 350 (the maximum supported by the device).

Message recitation is always interruptible, as is all Phoneshell output. In addition to skipping forward and backward between messages, the user may delete the message, request more information, send a fax copy of the message, or generate a reply. “More info” gives the full network address of the sender, a list of all other recipients of the message, and the time of message delivery in terse approximation such as “About five minutes ago”, “Early yesterday morning”, or “Last Thursday”. The fax option is useful for messages which require careful review or are difficult to comprehend when spoken, such as a detailed schedule and agenda of a proposed visit.

Both text and voice replies can be sent in response to a message. Sending a text reply requires typing the message with two touch tones per letter; the first chooses the alphabetic triplet associated with each key (for example, 3 maps to D, E, or F)[4] and the second tone selects one letter from the three. Although this sounds tedious, it is quite adequate for very short responses. A voice reply is recorded and the user is prompted to select a message format for the multimedia email reply; support is provided for Sun, NeXT, and MIME formats, as well as generic uuencoding and Apple AIFF files in a bin-hex form. The caller can edit a Cc list to send the reply to all or only some of the recipients of the original message, or add additional recipients selected from the caller’s own rolodex or the list of all local Phoneshell users.

Calendar

The calendar application under Phoneshell allows users to review and add entries to their personal calendars. While listening to the entries for a particular day, the user can jump to the previous or next day, or the next week, or choose a different date by specifying the day and month with touch tones. Calendar entries may be text or voice, and are synthesized or played accordingly.

In addition to the detailed presentation, increasingly terse summarization can be invoked with “week at a glance” and “month at a glance” functions. These examine the calendar for entries which either are repeated regularly, span multiple days, or contain key words. For example, a week might be summarized as “You have the usual meetings Monday and Tuesday, an an important meeting Monday afternoon. Nothing scheduled Wednesday. Thursday and Friday you are in Palo Alto.” The summarized format is more convenient for the task of scanning to select a date for an activity which will occur more than several weeks away.

To add an entry to the calendar, the user picks a date and records a voice message for that day. A special function lets the user add a fax number to the calendar for a range of days; Phoneshell and various other applications consult the calendar whenever a fax is to be transmitted, so the user does not have to enter the number for every request. The user can also request a facsimile calendar summary for the current month.

Rolodex

Phoneshell’s personal name and address application allows an individual to access information such as addresses and phone numbers, place onward phone calls, and send email (voice or text); the caller can even create a new rolodex “card” using touch tones. A card is selected by partially spelling a name, using one touch tone per character. Constraints based on the limited number of names in this database allow most names to be uniquely specified within three or four tones; in any event, a means must be provided for choosing between conflicting names, e.g. John Smith and Jane Smith (see [3] for a discussion of the disambiguation of touch tone letter triplets for spelling). The caller can optionally choose an alternate search criteria, based on first name, company name, or computer login ID, instead of the default last name search criterion.

Once selected, various fields in the card can be recited using speech synthesis; these include home and work telephone numbers and postal addresses, email address, fax number, and a field for optional remarks. Specialized routines translate some of these fields (address, phone numbers, and email addresses) into an intermediate form for speech synthesis. This generates much more intelligible synthesis; for example my telephone number (253-5207) is pronounced “two five three, five two oh seven” instead of “two hundred fifty three hyphen five thousand two hundred seven”, and the periods in an Internet mailing address are pronounced “dot”.

Additional information can be stored for any card in a list of “link” files. A link may be ASCII text or a Postscript image,

[4] In North America, the telephone keypad displays the letters on the telephone dial. This harkens back to the days where exchanges bore names, resulting in numbers such as “EDgewater4-8648”.

and might contain information such as driving directions or a map. Text links can be spoken, and either text or image links can be faxed to the caller. Links also provide a convenient facility for the Phoneshell user to store information (again, most likely a map) which one is likely to wish to fax to others; by storing these files under one's own rolodex card they can be sent anywhere, as Phoneshell prompts for a fax number destination.

Other action can be taken once a card is selected; most useful is that the user can place an onward call to the selected party. Phoneshell actually places a conference call to the desired party and waits several minutes listening for a touch tone escape sequence; if the user encounters either a busy or no answer condition, or leaves a short voice mail message, this tone sequence terminates the onward call and returns to Phoneshell. The Phoneshell user can also send a voice or text email message to the selected party exactly as is done in the email application.

Dial-by-name

An additional phone dialing application provides dial by name from a lab-wide database of about 50 names. This was provided as a separate application for several reasons: First, this is a public database, maintained for the entire Phoneshell community of users, and thus distinct from the personal rolodex. Second, the dial-by-name application also runs in a stand-alone mode on a different phone line, allowing anyone to phone into the lab without requiring an account and password under Phoneshell.

News services

News, traffic, and weather information are also available. News and traffic are periodically digitized from local radio stations. News is the hourly five minute broadcast on National Public Radio. Traffic reports are recorded every 10 minutes (the timing is approximate, so several minutes are recorded); because each traffic report is not complete, the most recent hour's worth of traffic files are saved and the user can jump between them. Recently the digital audio programs produced by Malamud's Internet Talk Radio have also been made available through the news application.

Local and national weather can be synthesized or faxed. NOAA weather information, including current conditions, near and extended forecasts are available in text form over the Internet from several hosts. Phoneshell maintains a menu of available cities and their associated three letter weather station codes. These forecasts are more succinct and often contain more timely information, such as detailed bad weather advisories, than can be obtained from the local weather recording or commercial telephone-based national weather services.

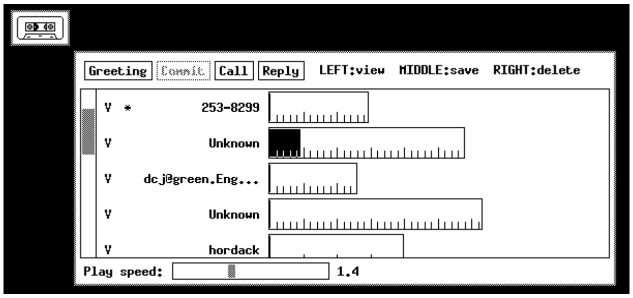


Figure 3: The screen interface to voice mail messages. Note that the third message arrived as email over the Internet.

Personal locator

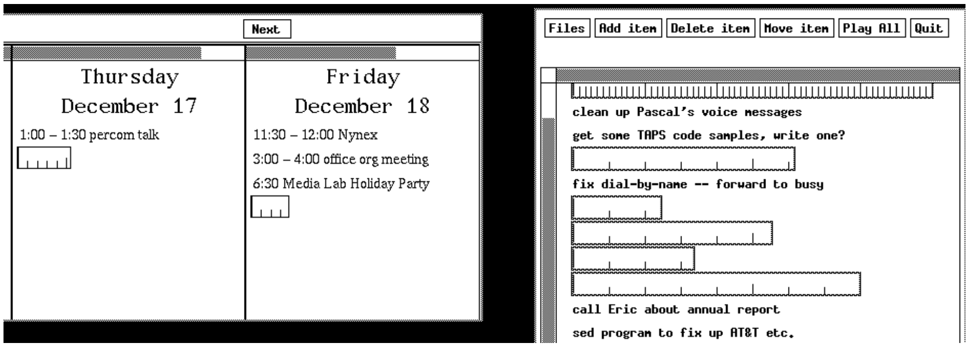
The final Phoneshell application provides a means of finding out who is in the lab, or when a user was last sighted, and can call someone in the lab on the nearest telephone. People's locations are gathered from Active Badges and Unix "finger" information, and summarized with speech synthesis. Calls are placed using the conference call mechanism described earlier.

DESKTOP APPLICATIONS OF STORED VOICE

This section briefly considers a number of screen-based (X window system) applications that support graphical and text access to the same databases available by Phoneshell. Some of these applications make no use of stored voice and are not described here; the user interface to the rolodex and speed dialing was described in [13] and the visual interface to personal location was described in [16]. Other applications are exclusively for voice; a voice mail viewer provides access to telephone messages and a speech-oriented audio editor supports other audio applications. Perhaps the most interesting applications are those which are logical counterparts for existing applications which until now have been exclusively text-based.

The screen interface to voice mail, shown in figure 3, is used for access to all voice messages, either from telephone messages or sent as voice annotation to email.[5] The calling party's name or number is shown if available, and a user can place a return call or record a reply to such messages. Each message is displayed using a SoundViewer widget. This widget is given a file name by the application, and displays a horizontal bar, the length of which indicates message duration. As the message plays, the widget fills in from left to right in synchronization, and the cursor can be moved with the mouse to provide random access. A speed control at the bottom of the window controls playback rate; the SoundViewer widget also supports a playback rate resource which

[5] A mail filter detects voice in incoming email, and diverts it to the voice mail inbox.



Thursday
December 17
1:00 – 1:30 percom talk

Friday
December 18
11:30 – 12:00 Nynex
3:00 – 4:00 office org meeting
6:30 Media Lab Holiday Party

clean up Pascal's voice messages
get some TAPS code samples, write one?
fix dial-by-name -- forward to busy
call Eric about annual report
sed program to fix up AT&T etc.

Figure 4: Voice and text in a calendar (left) and project list (right).

can be changed dynamically by key presses.

The SoundViewer widget is used in the graphical user interfaces of all the applications employing stored voice. Figure 4 shows two applications which mix voice and text. On the left is a daily calendar, while the one one the right is a mixed voice and text "things to do" list. Audio can be recorded into either application via the graphical user interface, but is much more likely to be recorded over the telephone. SoundViewers support cut and paste actions as well, allowing a portion of any sound displayed on the screen to be moved into another application or the audio editor.

PHONESHELL USER EXPERIENCE

Phoneshell was intended to support mobile users, by providing *convenient* access from the ubiquitous telephone; its users call in from cellular phones, from pay phones in airports, hotels, and even by the road in the middle of the desert, from phones in airplanes and overseas, as well as from home phones or even while visiting another person's office in the same building. This section describes what we have learned from these experiences. This is by no means meant to be a formal user study, much less a market analysis; many Phoneshell users helped develop it, and the majority of them are students who use only a subset of its capabilities. As a further caution, the author notes that he is the most avid user, and accounts for approximately a quarter of all the user experience to date. Nonetheless, this experience helps demonstrate situations under which remote access is invaluable and how its variety of functionality weaves a coherent whole.

Although Phoneshell remains under continual development as new applications and functionality are added, it has been running in some form continuously for over two years. It currently is running on three phone lines in two locations (the Media Laboratory and Sun Microsystems, which sponsored much of the work); at MIT two Phoneshell lines have been established to meet user demand and the need for reliability. At MIT, Phoneshell is called an average of eight times per day throughout the week. The voice mail server records approximately twenty messages a day during the work week, with fewer calls during the weekend.

Across these two sites, two "power users" make frequent access to the full range of Phoneshell functionality. One group secretary at MIT makes routine use of the screen-based applications for stored voice, managing four voice mailboxes at once, but calls in only for normal voice mail functions. About a dozen undergraduate and graduate students call in on average every two days, mostly for reading email; graduate students also receive voice mail from calls to their office phones, but undergraduates receive voice messages only from other Phoneshell users. Additionally, three other Media Lab research staff members use the voice mail capability without bothering with the Phoneshell interface; one of these users receives all his voice messages as NeXT multimedia mail forwarded to his workstation, another relies exclusively on the group secretary, while a third uses a mix of these strategies.

Phoneshell users clearly fall into two categories. The majority of users call in for messages, either voice or text, and do not use other features; in retrospect this should have been expected, as students do not keep large rolodexes or deal with a significant amount of urgent business correspondence, and have fairly predictable daily schedules. The two power users spend much more time working away from the office and routinely utilize the full spectrum of Phoneshell features, much as described in the opening scenario. Phoneshell has decreased the latency with which group members respond to messages (at least in their correspondence with the author, who is their supervisor) because telephone access is most likely to be employed when one is away from the laboratory for some time.

The dominant issue in Phoneshell usage is its ability to deliver information concisely with a minimum of user interface overhead; this includes speaking rapidly, speaking concisely, and always allowing the user to interrupt. Secondary factors have been its availability (reliability) and hence ability to perform in a timely manner, and the degree to which Phoneshell fits in with the work styles of its user community, wherein its integration with other desktop applications is especially salient.

Several factors contribute to the terseness of Phoneshell interactions. Users can and do increase the rate of both synthesized and digitized speech; listeners quickly adapt to faster speech [8], and, after some exposure, find normal speech rates unseemingly slow [1]. Filtering at many levels is essential. Some filtering is automatic, such as skipping over most fields in email headers, summarizing a calendar by week, and simplifying the presentation of time. Other filtering, particularly email sorting, is under control of the user, and all frequent users employ it. Listening to email is in no way a replacement for reading it, and isn't practical to listen to all of the 50 to 100 mail messages a day many of us receive.

Because it is designed for frequent users, the user interface to Phoneshell is more terse than many IVR systems for the general public. In addition, all prompts are interruptible, (this is enforced by the toolkit), making it possible to "type ahead" to skip menu prompts or abort playback of an uninteresting message. Combined with filtering and speech speedup, these factors alleviate at least some of the frustration we have all experienced dealing with telephone menus.

As previously noted in Nicholson's study of users of an integrated voice and text mail system [6], Phoneshell users tend to choose an appropriate medium for replying to messages, weighted by the medium which the sender employed and the importance of a quick response. In a community where text mail already dominated, voice messages have not replaced text, but they are used freely to reply to urgent text messages while out of the office. But as with any new application, availability and reliability have been important; while Phoneshell was under early development and crashed or failed to answer the phone nearly as often as it performed effectively, few users bothered to try it. More important, individuals tended to avoid voice replies to messages until the whole group was using voice. In many ways screen-based voice mail, which runs independently of Phoneshell, paved the way for wider use of voice across all applications. But now new students and employees readily accept that Phoneshell is part of our daily work environment.

In terms of specific applications, Phoneshell's version of voice mail offers little over most commercial voice mail systems; it has left many users convinced that listening to telephone messages without speech rate modification is quite tedious, however. More interesting are the new possibilities arising from the screen-based interface. Users are able to save and manage many more previously read messages in their voice mailboxes; because of the linear access styles of telephone based voice mail this is generally not possible without a visual user interface. The screen-based interface has made it possible for our group secretary to manage four voice mailboxes relatively easily; if only the telephone-based interface were available she would be endlessly polling these mailboxes with repeated phone calls.

As already mentioned, email filtering is essential, and the simple filtering based on sender and subject works surprisingly well. Its main failure is that it cannot yet adapt to changes in user habits, or know, for example, that if the author is visiting Palo Alto and has a meeting scheduled at PARC that email from xerox.com might be more important than it would be otherwise. Sending voice replies to email works quite well, though many recipients are surprised by having to use the multimedia extensions of their email systems for the first time. Typing short messages with touch tones is feasible and certainly better than having no means of replying to an urgent message, but much too tedious for more than several sentences. It is essential that the mail reader operate on the user's real mailbox, and not a copy; if messages cannot be deleted or marked as "read" when they are heard, much time is wasted cleaning up later.

For the rolodex, text processing was essential for making the various fields of a card intelligible. The dominant use of the rolodex over the telephone is for placing onward calls or sending voice messages. Links are a new feature which have interesting potential, but have not been used much in real life yet, although the author frequently faxes himself rolodex cards while traveling. Accessing addresses over the telephone has not proven to be a great benefit.

PHONESHELL ARCHITECTURE AND TOOLKIT

This section describes two aspects of Phoneshell's software environment: the relationship between Phoneshell and other applications and their underlying databases, and a toolkit for managing telephone-based audio user interaction.

Architecture

As Phoneshell was developed, it was intended to interoperate as much as possible with both existing desktop applications, which were based largely on text interfaces, as well as new graphical applications which would be required to access multimedia (i.e., voice) entries from the newly enhanced databases. As shown in figure 5, underlying databases can be presented aurally (by Phoneshell), graphically (by the screen-based applications) or as text, (e.g., by text editors or existing mail readers); the presentation medium dictates the interaction style and any necessary media translation, (e.g., from text to speech). The underlying databases were extended to include both voice and text. Other Unix processes, such as mail daemons, update these databases from time to time with appropriate file locking.

The new audio-capable applications were made to utilize existing database formats to whatever extent this was possible, and all databases are ASCII as they are accessed from a

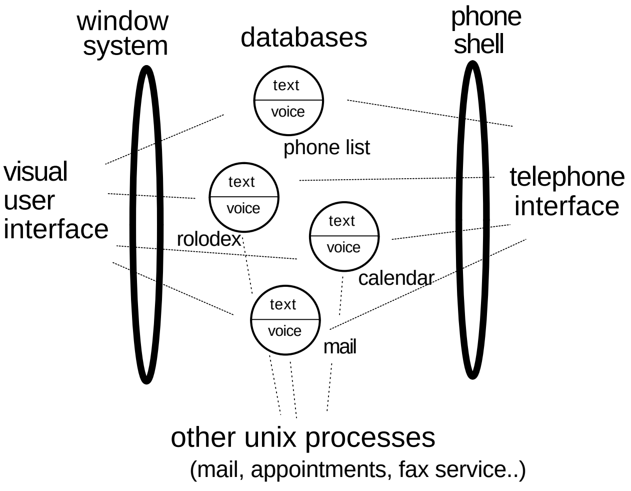


Figure 5: Multimedia databases are presented using different media and by multiple applications.

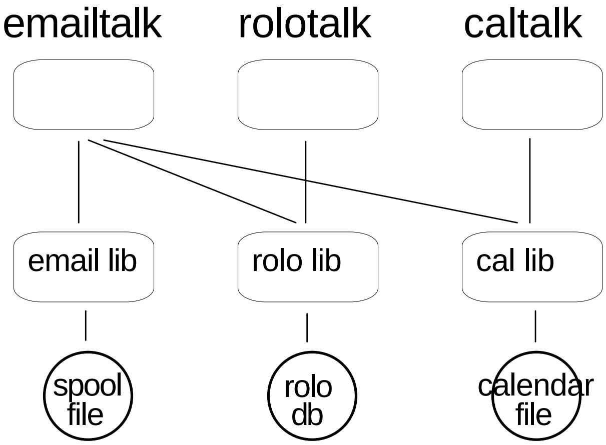


Figure 6: Applications embody user interfaces, and access their underlying databases through an intermediate "format independent" layer. Some applications require access to other applications' databases as well.

number of machine architectures with differing byte orders. For example, since Phoneshell users were already employing three different mail readers, each of which stored messages in a different format, Phoneshell needed to understand three formats. Phoneshell's calendar database was made to be a superset of the existing simple Unix calendar utility, with the addition of times, events which spanned multiple days, events occurring on "every Thursday", and audio data. In the calendar database a voice entry is stored as a file name; intermediate presentation routines determine whether to synthesize or play it (for telephone presentations) or to use text or SoundViewer widgets for display (the visual interface).

For all the obvious reasons, applications were made to provide user interfaces, while intermediary libraries isolated them from direct knowledge about the formats of their databases. Several additional benefits accrued from this arrangements. First, changes to Phoneshell were very localized when it was adapted to work with different database formats, e.g., Sun's Calendar Manager. Second, many of the Phoneshell applications exhibit a high degree of interdependence. As figure 6 illustrates, each application accesses its primary database, but some must access other data as well. While reading email, for example, the user may wish to send a copy of a reply message to a third party found in the rolodex, or may wish to fax a copy of the message, which involves searching the calendar for a daily fax number entry.

Audio toolkit

Although it was not a primary goal of this project, during the course of implementing Phoneshell a fairly extensive audio interaction toolkit was developed. Audio interactors present data and handle user input much as do window system "widgets", but must additionally manage time. For example, audio playback requires time, a portion of a sound may be played repeatedly, playback may be interrupted by a touch tone, or the user interface may wish to pause for a specified duration after playback awaiting input. Consistent use of an audio toolkit provides for modular software development, and also helps create a uniform user interface across the variety of applications which employ the toolkit. This section briefly outlines the functions of the various Phoneshell toolkit entities; Resnick describes a similar toolkit in [9].

A **menu** presents the user with choices, using speech synthesis, and handles touch tone input, which is returned to the application. The application specifies which touch tones are valid input, the prompts that may be associated with them,[6] optional initial greeting messages on menu entry, and reprompt messages when the menu repeats its list of options. The application can also modify the default waiting time between prompt recitations and the number of times the list should be presented before the menu returns a "timeout".

A **list** allows the user to select one element from a list of text strings, such as while looking up a name from the rolodex or choosing a city weather forecast. The list supports completion of partially spelled names, and enables further selection when the user's input matches multiple names. The list is driven by a file specifying the valid list entries. A slightly modified version of the list is used for logins; an optional field in each line of the file specifies a password which must be matched for the list selection to succeed.

A **prompt** speaks a message and waits a specified time interval for a touch tone reply. Any tone immediately interrupts

[6] Some valid inputs may have no prompts. For example, if a single escape key is used throughout an application, the application may wish to not repeat its prompt with each menu.

speech output. A prompt is used in situations such as when the application speaks “Press any key to cancel.”

A **text reader** speaks paragraphs or files of text interactively. It breaks the paragraph into sentences, searches local lexicons for alternate pronunciations of each word, and speaks the text sentence by sentence. During output, the user may skip ahead to the next sentence, repeat the current sentence, or quit. More specialized versions of text readers are designed to speak specific classes of text strings, such as telephone numbers, time, and internet mail addresses. The mail message reader analyzes the text body of a message for included text passages or email headers, and uses the text reader for each paragraph. An optional entry in each user’s .phoneshellrc file controls the rate of speech synthesis.

A **text creator** composes a text passage from touch tone input, with two tones defining a character, and “*” for the space character. It manages user feedback by speaking each word when completed, or spelling out an incomplete word when the user pauses for longer than several seconds. The text creator also handles a spoken help function, punctuation, word wrap on 80 character lines, and automatic or manual capitalization. A **number** handles numeric input, e.g., of telephone numbers, echoing the input for feedback when entry is terminated with the “*” key. A number can optionally be primed with a default and special prompt, allowing interaction such as “Enter a fax number, or press star for your default number.”

Similar toolkit entities handle playing or recording digitized speech. A **sound player** speaks an optional prompt and then plays a sound file; during playback the user may skip ahead or “rewind”, or change the playback speed, (as with speech synthesis, an entry in each user’s .phoneshellrc file specifies the default playback speed). A **sound recorder** speaks a prompt, records a response, and then plays the response back for the user to accept, cancel, or record anew. Recording is terminated when a pause or touch tone is detected; tone termination is useful when recording on especially noisy cellular connections.

CONCLUSION

Phoneshell was developed to take advantage of the ubiquitous telephone to provide remote access to timely information for mobile users. It has proven most successful for its most mobile users, and for at least several of them it has significantly changed the role of their desktop workstations while traveling; certainly a key to this success is Phoneshell’s ability to integrate many functions into a single interface accessible by a single phone call.

Visual user interfaces to stored speech are essential for realizing the full potential of voice for messages and as a data type in other applications which are accessed over the telephone. Telephone access gives new life to multimedia mail, and its value for particularly timely interactions gets the foot in the door for more pervasive uses of audio in desktop computing.

ACKNOWLEDGMENTS

Many members of the Media Lab Speech Research Group contributed to all as aspects of the applications described or mentioned in this paper, including but not limited to: Mark Ackerman, Barry Arons, Lorne Berman, Debby Hindus, Angie Hinrichs, Chris Horner, Eric Ly, Lorin Jurow, Atty Mullins, Sheldon Pacotti, and Jordan Slott. Don Jackson of Sun Microsystems was responsible for the port to the Sun site, and made numerous valuable suggestions about Phoneshell functionality. Finally, this paper never would have made the submission deadline without Gayle Sherman’s help.

This work was supported by Sun Microsystems.

REFERENCES

1. D.S. Beasley and J.E. Maki. Time- and frequency-altered speech. In N.J. Lass, editor, *Contemporary Issues in Experimental Phonetics*, chapter 12, pages 419–458. Academic Press, 1976.

2. Barbara L. Chalfonte, Robert S. Fish, and Robert E. Kraut. Expressive richness: A comparison of speech and text as media for revision. In *Proceedings of the Conference on Computer Human Interaction*, pages 21–26. ACM, Apr 1991.

3. James Raymond Davis. Let your fingers do the spelling: Implicit disambiguation of words spelled with the telephone keypad. *Journal of The American Voice I/O Society*, 9:57–66, Mar 1991.

4. T. W. Malone, K. R. Grant, K-Y. Lai, R. Rao, and D. Rosenblitt. Semi-Structured messages are surprisingly useful for computer-supported coordination. *ACM Transactions on Office Information Systems*, 5(2):115–131, 1987.

5. Gale L. Martin. The utility of speech input in user-computer interfaces. *International Journal of Man/Machine Studies*, 30:355–375, 1989.

6. Robert T. Nicholson. Usage patterns in an integrated voice and data communications system. In *ACM Transactions on Office Information Systems*, volume 3, pages 307–314. ACM, july 1985.

7. Robert B. Ochsman and Alphonse Chapanis. The effects of 10 communication modes on the behavior of teams during co-operative problem-solving. *International Journal of Man/Machine Studies*, 6:579–619, 1974.

8. D.B. Orr, H.L. Friedman, and J.C. Williams. Trainability of listening comprehension of speeded discourse. *Journal of Educational Psychology*, 56:148–156, 1965.

9. Paul Resnick. Hypervoice: A phone-based CSCW platform. In *Proceedings of the Conference on Computer-Supported Cooperative Work*, pages 218–225, New York, Nov 1992. ACM.

10. Marshall T. Rose. *The Internet Message*, chapter 6. Prentice Hall Series in Innovative Technology. P T R Prentice Hall, Englewood Cliffs, NJ, 1993.

11. S. Roucos and A.M. Wilgus. High quality time-scale modifications for speech. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 493–496. IEEE, 1985.

12. Christopher Schmandt, Mark S. Ackerman, and Debby Hindus. Augmenting a window system with speech input. *IEEE Computer*, 23(8):50–56, Aug 1990.

13. Christopher Schmandt and Stephen Casner. Phonetool: Integrating telephones and workstations. In *Proceedings of the IEEE Global Telecommunications Conference*, pages 970–974. IEEE Communications Society, Nov 1989.

14. Lisa J. Stifelman. Not just another voice mail system. In *Proceedings of the 1991 Conference*, pages 21–26. American Voice I/O Society, Sept 1991.

15. Lisa J. Stifelman, Barry Arons, Chris Schmandt, and Eric A. Hulteen. VoiceNotes: A speech interface for a hand-held voice notetaker. In *Proceedings of INTERCHI '93*, New York, Apr 1993. ACM.

16. Steven Tufty. Watcher. S.B. Thesis, MIT Dept. of EECS, 1990.