Caltalk: a Multi-Media Calendar

Chris Schmandt

Media Laboratory
M.I.T.
E15-327
20 Ames St.
Cambridge, MA 02139

617-253-5156
geek@media-lab.media.mit.edu

Introduction

This paper describes *caltalk*, a voice and telephone interface to the common Unix calendar program's database, a slightly formatted ASCII file. Caltalk allows users to dial in, hear entries in their calendar using text-to-speech synthesis, and add new entries to the calendar by digitally recording a message. The name of the resulting sound file is stored as distinctively marked text in the calendar database, allowing continued operation of the original calendar program.

We also built a graphical interface, *xcal,* to the same database. This tools provides window-based display of days, months, and years, much like an appointment book, and is in this respect similar to a number of other calendar programs. When run on a workstation which is audio-capable, however, this interface intercepts the encoded text strings which indicate sound files, and instead displays a SoundViewer, a graphical sound interactor, in the appropriate calendar slot.

Voice in the Calendar

A calendar is a challenging database, since it must be kept current to remain useful. Not only is the calendar used to store appointments, it is also consulted for scheduling purposes. A voice interface over the telephone can enhance the calendar by allowing greater *access* to information stored in it, using text-to-speech synthesis to describe appointments. This interface also allows improved *consistency*, in that an entry may be added to the database very soon after it is scheduled (using digitized speech) instead of waiting until returning to the office.

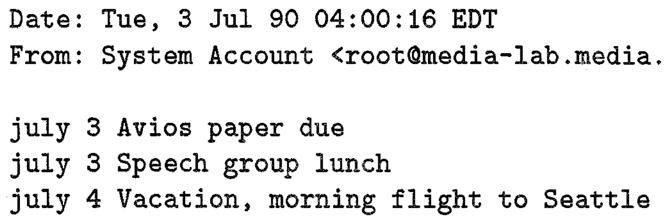
The Unix *calendar* program uses a text file as its database. No user interface is provided for adding or removing entries; this is done using any text editor. The file (see figure 1) associates month and day with a single line entry. The calendar program is run daily in the early morning, and sends the user an electronic mail message reminding of scheduled events for that day and the next (see figure 2). This is a very primitive interface, in contrast with to others which use bitmap displays, mice, and fancy graphics; yet it is attractive in that it is universal, and accessible from any terminal. The advantage of this calendar is not its interface, but simply maintaining the discipline of keeping everything in one database.

The voice interface of caltalk uses touch tones for input and speech synthesis for most audio output. A user calls up, logs in, and has access to a number of top level functions. These functions include selecting a day, reviewing the selected day, week at a "glance", or adding a calendar item. Operations apply to the "current day", which may be specified by choosing a month and date, and defaults to today.

The "week at a glance" function (touch tone 4) scans the week in which the specified day occurs, starting from Monday. If the day is within the current week, it simply scans to the end of the week. This entails repeated subroutine calls to a function recites the schedule for a single day. This

```
july 2 Media Lab forum
july 2 Kaya's birthday
july 3 <sound_file: cal.032>
```

Figure 1: Calendar file format. The entry for 10 November was added by *caltalk*.



```
Date: Tue, 3 Jul 90 04:00:16 EDT
From: System Account <root@media-lab.media.mit.edu>

july 3 Avios paper due
july 3 Speech group lunch
july 4 Vacation, morning flight to Seattle
```

Figure 2: The mail message delivered by the calendar program on 23 October.

function is also available directly from the top level (by pressing 3).

During review of a day, the calendar entries are presented sequentially. A text entry is recited by the speech synthesizer. A voice entry is played back. In either case, there is a slight pause between entries. At any point during presentation, the user may hit a key to repeat the entry, to jump to the next or previous entry or day, to delete the entry, or to add a new entry.

An entry may also be added directly from the top level (by pressing 2). To add an entry, the user selects a date, using touch tones for the month and year, and is then prompted to start recording. A short sample session follows. This script assumes the user has already logged in.

"Top level. Monday, November 6. To select a date, press 1. To add an entry, press 2. To review a day press 3. For week at a glance, press 4. You may always press O for help"

user presses 3 (`review current day`).

"Review day. Speech class 10 AM" (pause) "12:30 Lunch meeting with Sheldon and Lorne"

user presses 6 (`repeat`).

"12:30 Lunch meeting with..."

user presses 2 (`add entry`)

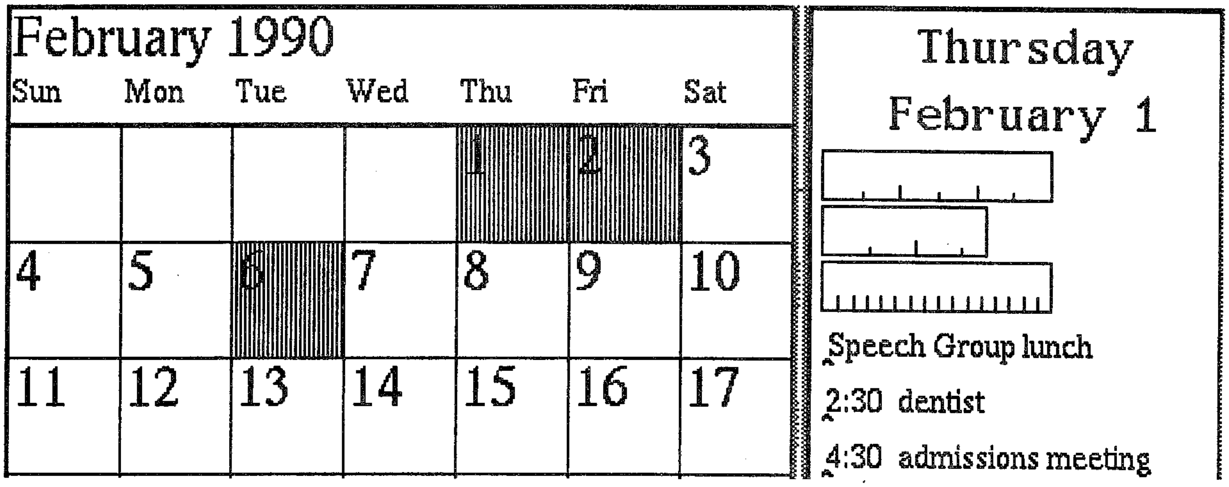
"Record your entry now, or hit any key to abort." (starts recording)

The top level entries are organized to provide convenient access to the most common functions, i.e., calling in to add an entry to the calendar, or to scan a day or week for scheduling an activity. Although it is relatively easy to find out what I am doing "today" (one keystroke from the top level menu selects review a day, and the day defaults to today), this is not expected to be the primary use of caltalk so it was not made the default.

Design Considerations

Caltalk seeks to provide a increased calendar access using speech output. Speech is a particularly difficult medium, and suffers from slow speed, a serial nature, and for synthetic speech, poor intelligibility with a high cognitive load on the listener. Voice systems must be designed carefully to compensate for these limitations.

Navigation among voice menus is particularly difficult because of the cognitive load on the user. Caltalk menus always identify themselves, (each menu has a name) to develop some sense of place.



February 1990

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17

Thursday
February 1

Speech Group lunch
2:30 dentist
4:30 admissions meeting

Figure 3: The visual calendar, showing a month view and a day view.

At no point are more than five choices presented to the user. Most keys map to single functions, and are always active. Context sensitive help is available at any time by pressing the "O" key. A geographic key layout is attempted; while listening to an entry, keys 4 and 5 are used for next and previous *entry*, while keys 7 and 8 (immediately below) choose next and previous *day*.

Because speech is slow, we need to minimize the time it takes to perform a transaction. All speech output is interruptible; this allows an experienced user to type ahead and avoid listening to menu prompts, which are necessary for novices. The menu hierarchy is shallow, and the top level is almost always one key press away. As mentioned above, important functions are accessible in many ways. For example, one may choose to add a calendar entry either from the top level (while reviewing a single day), or while using the "week at a glance" function.

Graphical Interface

The standard Unix calendar program provides a crude, character-oriented calendar interface, useful on any ASCII terminal. If caltalk has recorded voice entries, the calendar can present them only as the tagged text strings encapsulating a sound file name (figure 2 shows an example). Caltalk provides a remote audio-only interface for telephone interaction. It is equally important, however, to support a direct manipulation interface from the workstation window system; this may get much more frequent use than the telephone interface.

Xcal is a graphical calendar interface which runs under the X window system. It provides a visual interface for a year, a month, or a day (see fig. 3). Mouse and keyboard can be used for selecting, displaying, and entering calendar entries. In this sense, Xcal is similar to many other window-based calendar programs; however, these programs generally ignore the existing calendar program and invent a new database format. This makes the calendar inaccessible from dumb text-only terminals.

Xcal displays sounds graphically, using a *SoundViewer widget* (figure 4).[1] The SoundViewer at-

[1] *Widget* is the X term for a graphical user interaction object, such as a scroll bar, toggle button, etc.

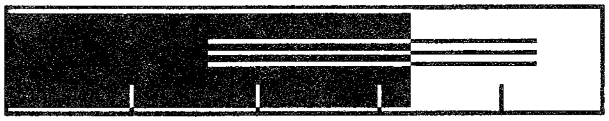


Figure 4: The SoundViewer widget. Vertical marks are ticks indicating time. The black bar shows the current sound position. The horizontal bars show the extent of the currently selected sound.

tempts to use spatial context to provide cues for the temporal nature of speech. As the sound plays, the bar cursor moves in sync. Tick marks provide visual cues for units of time. The sound cursor may be grabbed, moved, and released with the mouse, to repeat or jump ahead during playback. A portion of the sound may be selected and deposited into another sound-capable application, just as text may be moved between windows. The SoundViewer is used as the single consistent sound user interface in all our graphics applications.

Coexistence of Media

Viewed together, these two calendar interfaces illustrate some important principals in the design of multi-media systems:

- The need for a coherent rationale for introduction of new media.
- The importance of interoperability with existing tools, achieved by translation of data types when possible.
- Scalable applications, which can provide increased functionality for more media-capable systems, but still operate on less powerful workstations.
- Potential benefit from the consistent use of new data types across applications, including user interface.

These may be the most salient points of this project for multi-media workstations.

Conclusions

This paper has described a working prototype of a multi-media calendar, with voice and graphical interfaces, and employing recorded voice as a data type. Although the application is simple, it exhibits some important traits of multi-media systems in general. Particular emphasis has been placed on the need for interoperability with existing applications, other media.

Acknowledgments

Sheldon Pacotti wrote *xcal* and the data abstractions for *caltalk*. Mark Ackerman wrote the SoundViewer widget. This work was supported by AT&T and Sun Microsystems, Inc. Elaine McCarthy helped edit this paper.