

What does Pink sound like?

Designing the Audio Interface for the TalOS

[Tom Dougherty, Ph.D., Interval Research Corporation](#)

To Pauline Oliveros, who encourages me to always listen to everything.

This paper discusses the design considerations of an auditory interface for an operating system with a graphic user interface. The Cutting Edge Scenario, the last publicly shown Taligent OS demo, illustrates certain audio interface design points. Additional subsequent ideas regarding adding audio to an operating system like the TalOS will be discussed.

History

In early 1986, a team in Apple Computer's Advanced Technology Group began a research project that was code-named Pink, with the goal to create an operating system for the next generation of computers. In 1990, the project became TalOS—the property of Taligent—a joint venture between Apple Computer and International Business Machines. In 1994, Pink's OS dreams died as Taligent redirected its efforts toward CommonPoint™, a non-OS application environment which would exist as a layer on other operating systems. The demo that I will show at ICAD is the last publicly shown demonstration of the interface directions of the Pink operating system.

TalOS was unique in its architecture. It was object oriented from the kernel up, and provided true pre-emptive multi-threaded multi-tasking. The end user experience revolved around a compound document-centric, multi-user networked, direct manipulation interface with infinite session undo. The principal interface theme was People, Places and Things. The

networked interface represented remote users, as well as collaborative work spaces. In many ways it was more a graphic MOO (multi-user dimension-object oriented) than an traditional operating system.

Audio Interfaces in Operating Systems

Auditory icons have been discussed for years, although few commercial products employ them. My goal as the designer of the TalOS audio interface was to design a high quality auditory experience that could be informative, comprehensible and enjoyable. Pink's auditory interface integrates many of the concepts of prior work in this area, including Gaver's (1989) work on the Sonic Finder as well as Cohen's (1994) work on monitoring background activities.

Why Add Sound to the TalOS?

I began designing an auditory interface for the TalOS in order to extend the real world metaphor of the interface. An auditory interface can support the decentralization of feedback and status by displaying sounds instead of always displaying modal dialog boxes. It also enhanced the physics of the interface objects (sonifying drag acceptance feedback to indicate how the target; e.g., folder or printer, operates on the item being dragged to it; e.g., file. Finally, interface audio helps create cues useful in navigation-giving each Place its own ambient sound based on the People, and Things contained within.

One possible reason for the lack of ubiquitous auditory interfaces in commercial operating systems is the lack of a supporting architecture. The infrastructure required to sonify an operating system in a meaningful way was available in the TalOS. It included true multi-threaded multi-tasking, and object orientation, for extensibility. The media frameworks allowed for integrated telephony, real-time mixing of multi-channel audio, support for 16 bit, stereo audio, MIDI

support, and a master clock with absolute time for time-based media synchronization.

Audio Interface Design Goals

The principal goal was to present meaningful audio support for graphic interface. In order to create feedback that people would use, I realized that it would need to be of extremely high quality (16 bit stereo) and extremely low volume. My volume yardstick was that most audio should not be much louder than the typing on the keyboard. Another requirement was that the user can always manually turn off sound, and that certain applications could also disable the audio feedback (e.g., audio recording).

Political struggles

Some engineers understood importance of good sound design in the interface, but others were concerned that their computers would sound like a video game. Many had used the Sonic Finder when working at Apple, and were concerned about a similar sound quality (8 bit) and volume. They had to be assured that the volume would be extremely low and that the sounds would be of compact disc quality. They were also assured that there would be a way to disable classes of sound interface elements without turning off all audio.

Learning the meaning of the sounds

Cohen's ShareMon ([1994](#)) presented users with dynamic file sharing information. While he reports that some users found it useful and successful, his usability analysis suggested that users often had a difficult time interpreting the meaning of many of the sounds.

In a true multi-tasking operating system, managing background tasks is critical. In TalOS, most processes possess associated interactable interface objects. For example, users may initiate a print job by dragging a document (or its proxy) to a visual printer appliance

icon. Upon acceptance, the printer animates, and audio is heard to indicate the status of that job. In this example, the printing is a foreground process, or the current user focus. However, when the user changes her focus to any other activity which occludes the visual representation of the printer object, printing becomes a background process. Because the user directly experienced the printer initially, and heard the associated audio, she learns specific interface sounds by association with the appliances which embody those processes. When audio of a background task is presented to the user, the user recognizes the sound without the visual indication since she had learned to associate that sound in the previous foreground task.

Sound Design and Mapping

The real world is a semantically rich place to start to look for creating appropriate mappings between interactions, interface objects, and sound. The interface sounds were 16 bit, 44.1 kHz samples that were recorded in-house, and synthesized or taken from public domain sound effects libraries.

Why use sampled audio?

While there are some good synthesis algorithms for creating parameterized auditory icons ([Gaver, 1993](#)), not enough synthesis algorithms accurately support an entire taxonomy of interface items necessary for our purposes. With the right tools, manipulation techniques and mixing are sufficient to parameterize for most situations.

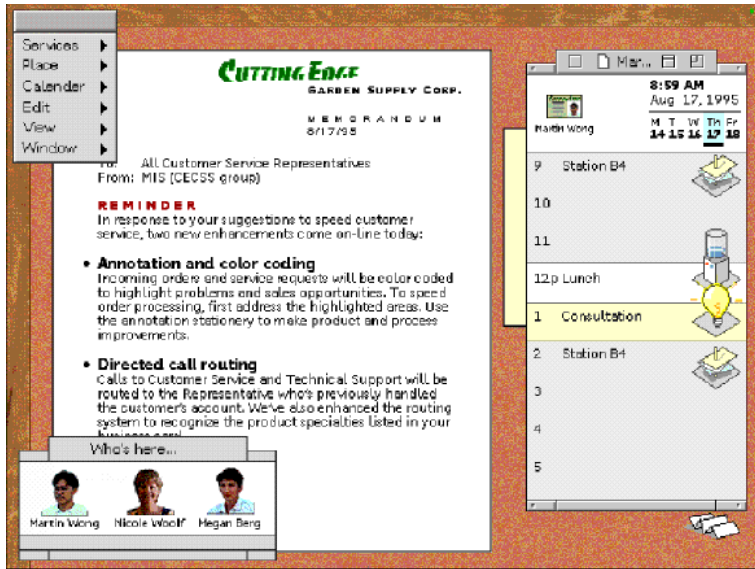


Figure 1. A screen from The Cutting Edge Scenario

Feedback

The Cutting Edge Demo

The demonstration (Figure 1) was shown to users with or without the interface sounds. While not mentioning the sounds specifically, those presented with the sonic version responded much more favorably, leading us to believe that the sound design was successful. In fact, their success was almost problematic since many of the users seemed influenced by the sounds although they didn't even notice them.

Subsequent Ideas and Techniques

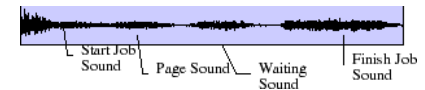
The following are additional techniques that I created after leaving Taligent based on extending the audio interface for an operating system like TalOS.

Multiple loop points facilitate parameterization

Sampled sound files can be parameterized when created with multiple loop points. Below is a diagram which describes how the sampled sounds could be prepared. The sample contains multiple semantic components which can be played in a non-linear manner depending on the state of the interface item.

Consider a user dragging a document to the printer. In this case it is a shared printer and there are other jobs to print in queue. The user would first hear the initial sound of the printer starting up, and then instead of proceeding to the printing loop, it would proceed to the wait loop until the user's job was ready to be printed. The printer would then loop for each page printed, and when complete, it would seek to the end loop.

Printer Sample Example:



Sound manipulation techniques

Stereo Panning

Stereo panning was used to cue the user to the location of sound emitting elements in the interface. In the visual attention literature, audio priming has been shown to decrease reaction time in finding visual targets on a computer screen. By using stereo cueing, visual representations of elements could be rendered in a more subtle, less distracting way, yet still draw the user's attention when necessary.

Dynamic muffling as a spatial cue

Dynamic equalization can be a useful technique for representing auditory occlusion. It is often not sufficient for a user to hear an auditory icon; the user needs to be able to determine the location of the associated visual interface object in order to manipulate it accordingly. Stereo panning is useful in displaying information along the horizontal azimuth, and equalization, or filtering can be

useful in presenting information about the z axis.

We can easily identify the sound of a passing car whether we are on the street or in our house. Although the spectral energy is considerably different, they both are heard as cars, even though the walls of the house muffle the sound of the car outside. Muffling is an auditory manipulation that can present additional information without compromising the identification of the source.

Hearing the current context

An operating system structured around compound documents provides many potential contexts for presenting auditory icons and processes. The hierarchy of user focus in the TalOS for a given task might be

- 1) the Place™,
- 2) the active, open document,
- 3) the visible page, and
- 4) the active frame.

The current frame is the user's current focus, and determines which tools are presented automatically for use.

Imagine that a frame requires attention somewhere. Perhaps the frame contains a chart which accesses data from the Internet on a periodic basis. If the status is not critical, a local sound and visual will be presented in place at that frame. If the sound source is in the active frame the audio will be presented normally. If the sound source is in an inactive frame the sound's high frequencies would be slightly attenuated through a minimal low pass filter. If the sound source is in the current document, but scrolled off the page, the sound will appear muffled. If the sound source is in an inactive but open document, its alert will be heavily muffled.

This muffling not only provides a spatial cue, but provides a mechanism for creating a richer acoustic environment. The monotony often associated with auditory interfaces is hearing the exact sound repeated, and this diminished as sounds are represented differently based on location and active context. The computer soundscape models the physical world in that distant sounds can be attended to if desired, but are quieter and subtler, making them less distracting. As the user travels through a

document space, the auditory environment adapts to display information according to the changing user focus.

Equal Loudness and Automatic Calibration

Two mechanisms were designed to assure that interface sounds were as unobtrusive as possible. They were dynamic equalization (above) and automatic level calibration.

Dynamic equalization was used to exaggerate the highest and lowest frequencies ranges for each interface sound. This process compensates for the equal loudness contour phenomenon. At low volumes, the high and low frequencies become more difficult to hear than midrange sounds. Because the sounds were originally designed to be presented at a very low volume, just above threshold, radical equalizing became necessary to insure that the character of sound elements was preserved.

Automatic level calibration

One design goal was to present auditory icons and status sounds just above threshold. Patterson (1982) sampled ambient sound levels in airplanes to determine the volume at which to display these alerts so as to be just above auditory threshold. Since the TalOS hardware assumptions required a microphone for telephony, I realized that the microphone could be used to *dynamically* sample the volume of the ambient workspace and compensate it. This way office noise wouldn't mask the interface sounds, and the sounds would be displayed at the lowest useful volume.

Summary

The design of an ubiquitous audio interface to an operating system should be unique to the specifics of that operating system. The designer must tailor the design to support the principal user tasks within the structure of that operating system. In the present example, the design space involved a compound document-centric, direct manipulation graphic user interface. These capabilities enabled several novel audio interface techniques described here. In this realm, the role of the audio is to present information to the user so that the context changes inherent in such a system.

Acknowledgments

Special thanks to Deborah Magid for permission to discuss and present this work, Jonathan Cohen and Harvey Lehtman for input on drafts of this paper, and to Steve Milne, Joy Mountford, Lyne Plamondon for their thoughtful discussions of sound. The Cutting Edge Scenario was created by the Taligent Human Interface Team: Kirk Scott, Jeremy Ashley, Tom Dougherty, Grace Colby, Sara Sazagari, Alex Liston, Lucy Berlin with Carl Stone and Keith Okabe.

References

Cohen, J. (1994). Monitoring Background Activities. In G. Kramer (Ed.), Auditory Display: Sonification, Audification, and Auditory Interfaces. Santa Fe Institute Studies in the Sciences of Complexity, Proceedings (Vol. XVIII). Reading, MA: Addison-Wesley.

Gaver, W. W. (1989). The Sonic Finder: An Interface that Uses Auditory Icons. Human Comp. Inter. 4(1).

Gaver, W. W. (1993). Synthesizing Auditory Icons. In Proceedings of INTERCHI '93 April 24-29, in Amsterdam. Reading, MA: ACM Press, Addison-Wesley.

Patterson, R. D. (1982). Guidelines for Auditory Warning Systems on Civil Aircraft. Paper No. 82017, Civil Aviation Authority: London.

Author

Tom Dougherty, PhD
Interval Research Corporation
1801 Page Mill Road, Building C
Palo Alto, CA 94304
(415) 842-6274
doughert@interval.com