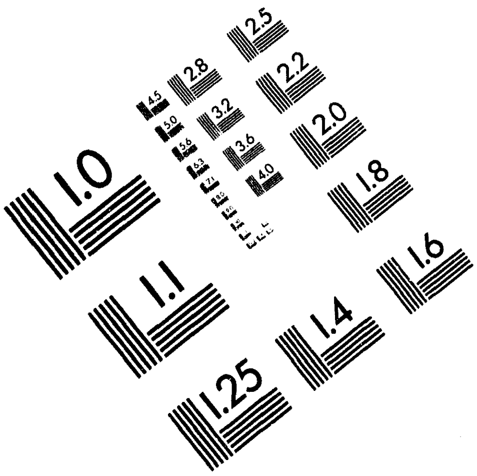
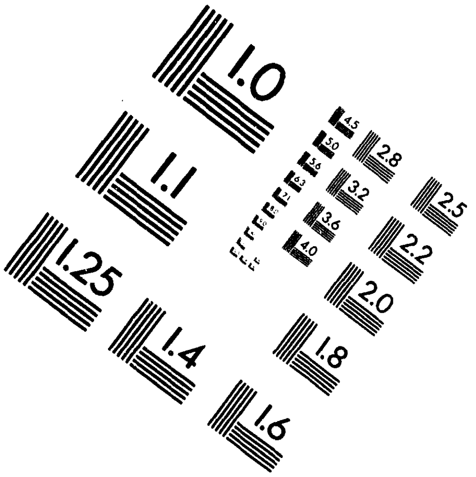


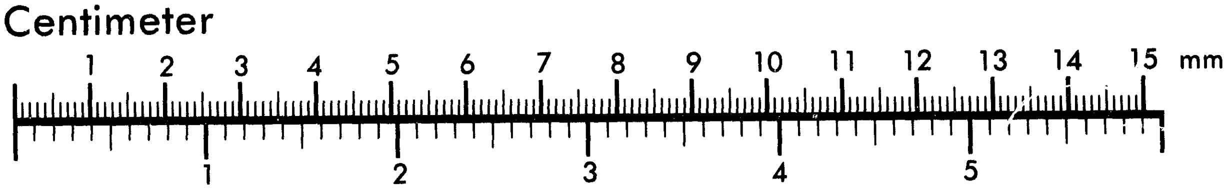
AIIM

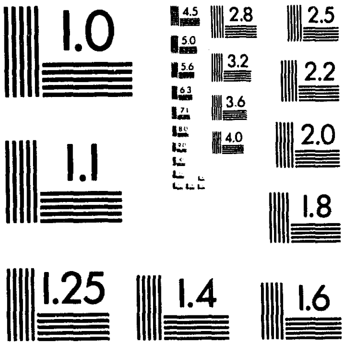
Association for Information and Image Management

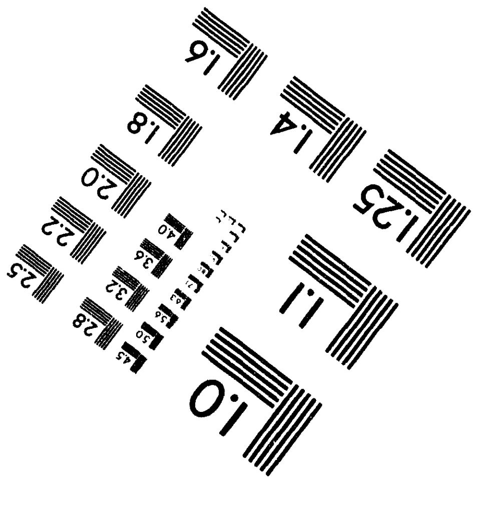
1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910

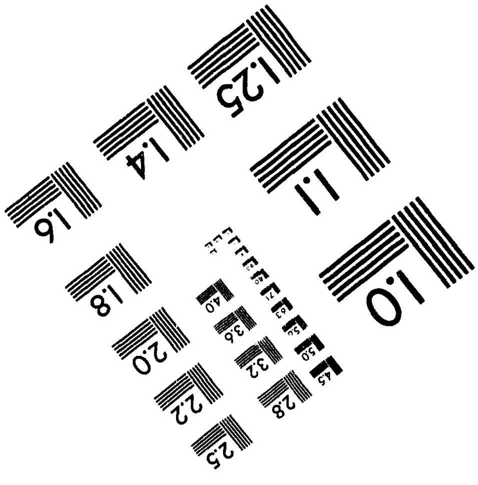
301/587-8202











MANUFACTURED TO AIIM STANDARDS
BY APPLIED IMAGE, INC.

1 of 1

A Centralized Audio Presentation Manager*

Albert L. Papp III
University of California, Davis, and
Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, CA 94551
Phone: (510)-423-9027 Fax: (510)-423-4139
E-mail: papp@llnl.gov

Meera M. Blattner †
University of California, Davis, and
Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, CA 94551
Phone: (510)-422-3503 Fax: (510)-423-4139
E-mail: blattner@llnl.gov

May 16, 1994

Abstract

The centralized audio presentation manager addresses the problems which occur when multiple programs running simultaneously attempt to use the audio output of a computer system. Time dependence of sound means that certain auditory messages must be scheduled simultaneously, which can lead to perceptual problems due to psychoacoustic phenomena. Furthermore, the combination of speech and nonspeech audio is examined; each presents its own problems of perceptibility in an acoustic environment composed of multiple auditory streams. The centralized audio presentation manager receives abstract parameterized message requests from the currently running programs, and attempts to create and present a sonic representation in the most perceptible manner through the use of a theoretically and empirically designed rule set.

Keywords: Speech and nonspeech integration, Information abstraction

*This work was performed with partial support of NSF Grant #IRI-9213823 and under auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

†Meera Blattner is also with the Department of Biomathematics, M.D. Anderson Cancer Research Hospital, University of Texas Medical Center, Houston.

1 Introduction

User interfaces which support concurrent program executions have little, if any, audio management. Typically, some number of audio channels exists as a resource, and programs request the number of audio channels required and the operating system either grants or denies the request. Therefore, in environments where multiple programs output sound, each individual program has no overall context of the auditory system state with the possible exception of how many audio channels the operating system has allocated for other applications. This potentially leads to numerous problems in the perception of the various sources of sound. Programs typically play audio without regard for the overall auditory environment which can cause sound masking and perceptual unintelligibility. When numerous applications output audio, there may be confusion in knowing which application produced what sounds.

These problems can be addressed by examination of the global auditory state of the computer system. This is realized through the use of a server through which all audio requests must be made. The auditory state is dynamic in that it is automatically altered depending upon the constantly changing auditory needs of the various applications running at the moment. Furthérmore, the user is able to set a number of parameters both globally and within each individual application to suit the audio interface to his or her liking.

The presentation manager receives descriptive messages which contain information about system activities and program states, as specified by the user or application programmer. The sonic output of the set of running programs and the overall auditory system state is controlled by the presentation manager. It chooses how the information is to be represented in sound, within the constraints of the descriptive message. The concept of dynamic representations of information in different forms, or *multimodal objects*, is discussed in [1]. The presentation manager must choose the form with consideration for other current sonic output. The appropriate auditory form must not only convey the informational content of the message but also keep all current audible information as perceptually clear and localized as possible. This is accomplished through the application of auditory streaming theory [2] and a psychoacoustic knowledge base.

In order to give the presentation manager as much flexibility as possible, applications needn't send raw auditory information. Rather, many common forms of data can be sent to the presentation manager along with more general information about that data. Depending upon the information received and the current auditory state of the system, the most appropriate auditory representation for the data will be used in its presentation. It is hoped that by using this method, the output will be perceptually clear and the developer will be saved from designing sounds which might conflict with the auditory output of other running applications.

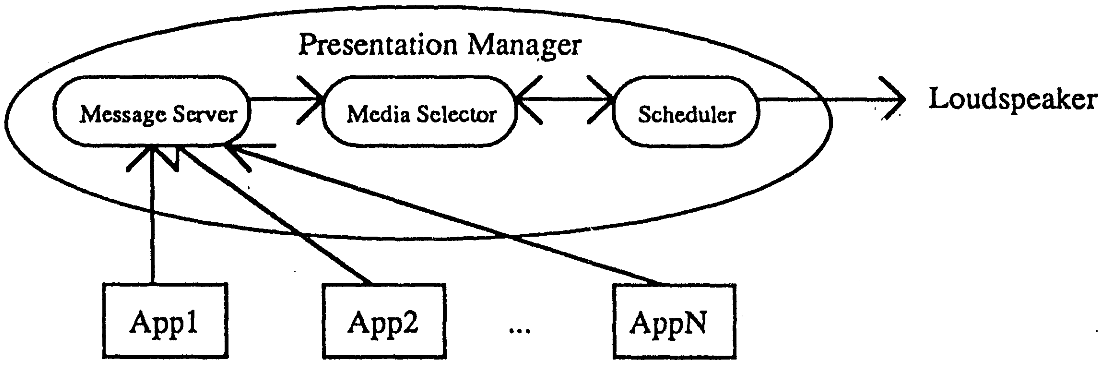


Figure 1: The Presentation Manager design

2 Presentation Manager Design

The audio presentation manager is composed of three distinct parts: the descriptive message server, the media selector, and the scheduler. Figure 1 serves to illustrate the overall system organization. A message passing paradigm serves as the underlying model for communication between applications and the various parts of presentation manager. The arrows indicate the flow of messages. The content of these descriptive messages and each of the parts of the presentation manager is described below.

2.1 Audio Scheduler Descriptive Messages

Whenever an application is to represent some information in sound, it sends a message to the presentation manager. This message includes a high level description of the information to be displayed. The presentation manager then decides *how* the message is to be displayed. This allows the applications programmer to concentrate on the application itself rather than its auditory presentation.

2.1.1 Descriptive Message Parameters

Descriptive message parameters describe the included information and help the presentation manager choose a suitable representation in sound. The range of values each parameter can accept is constrained initially by the applications programmer. The user can then alter certain parameter values to his or her liking.

Perception Style (*Persistent*, *Conditionally Persistent*, or *Discrete*)
This parameter indicates whether the message is presented persistently or only at specified times. For instance, one may wish to have a persistent message set to the number of users in a network accessing a file server. The parameter must present itself at all times. Conditionally persistent messages are persistent within some specified constraint. For example,

in the previous example, a conditionally pers'stent message could have the constraint that the number of users must be equal to or greater than three. Otherwise, the persistent message ceases. Discrete auditory messages present themselves only when a state has changed. For example, the state of a mailbox could be the number of unread messages in it. The change-of-state indicator will be triggered each time a new mail parcel arrives.

Urgency (*Low* **through** *High*) This parameter indicates the amount of urgency of the message. A high value indicates that it is very important the auditory message is clearly and hastily perceived over perhaps many other concurrently presented auditory messages. A low value means that the message is not of critical perceptual importance and its presentation can be put in the background.

Latency (*Zero* **though** *Infinity*) The latency value helps the scheduler make decisions of a temporal nature. Auditory messages which, for example, need to coincide with visual elements should have a zero value for latency. Messages which are not so time critical can have higher values. A latency value of infinity means that the auditory message is delayed until no other auditory message (with a value other than infinity) is waiting to be sounded.

Precision (*Low* **through** *High*) Certain descriptive messages must describe very precise information, such as an exact value or a specific instruction. The precision parameter indicates how exact the information must be. Typically, those messages requiring a high level of precision will be represented in voice, and those messages with lower precision will be represented using some form of nonspeech audio.

Information Type The information type gives the presentation manager additional information as to which forms of audio might be appropriate for this particular descriptive message. Information types are as follows:

1. Application State Indicator
2. Annotation
3. Event Marker or Feedback
4. Detailed Status Report
5. Numeric Value
6. Application Warning
7. Application Error

2.1.2 Global System Parameters

Global system parameters are those which do not have direct bearing on any particular application. Rather, they control some aspects of the overall interface.

Familiarity Level (*Low* **through** *High*) The presentation manager can alter the representation of an auditory message dependent upon the user's familiarity with the system. A lower familiarity level would cause the presentation manager to represent auditory messages sequentially, where possible, to avoid excessive overlap. The global parameters also change the interpretation of the descriptive message parameters. For the novice, (Familiarity level set to low) a high urgency message would be emphasized more than normal. Someone who is familiar with the interface would not need an amplified urgency level since that user would be more familiar with the auditory characteristics of an urgent message.

Ambient Noise Level (*Low* **through** *High*) Dependent on this parameter, the presentation manager will alter the volume and pitches of messages to make sure they are audible over the current background noise.

2.2 The Audio Message Server

The audio message server is responsible for receiving all auditory event requests from running applications. The global system information dynamic variables are read by the server and added to the messages before they are sent to the Media Selector. When the audio server receives a number of messages simultaneously, the order of their arrival at the media selector is determined from the ***urgency*** and *latency* fields of the messages.

2.3 The Media Selector

The media selector supports four specific auditory output media into which information can be encoded. Additionally, there is support for raw audio output as well. Each of these media works best to represent specific types of information. However, the decision of which auditory medium to use must take into account what audio is currently being sounded. After the media selector makes its decision as to how the information is to be represented, the actual audio data is sent to the scheduler. The work done by the media selector can be trivial, such as in the case that the incoming message is raw audio data. Since the media form is already predefined, the message is left unaltered and it sent out for scheduling. The media selector might also have to do a considerable amount of work, such as in the more abstract case of a message which contains a real number to be displayed. Based upon the values in the message fields (i.e., perception style, urgency, latency, ambient noise level, etc.) it must decide if that number should be spoken or represented through one of the nonspeech audio forms, using either some predefined numeric definition or perhaps some other more intuitive form. The supported auditory forms are described below.

2.3.1 Abstract Earcons [3]

Abstract earcons are short, distinctive audio patterns to which arbitrary definitions are assigned. They can be transformed in various ways to assume different, but related meanings. Using these principles, families of earcons can be defined. The complexity or simplicity of the earcons is controlled by the creator of the earcons. Abstract earcons share many advantages and disadvantages of spoken language. The arbitrarily assigned definitions require the user to learn the mapping between the sound and its meaning. With careful design, the learning time is minimal [4, 5]. Abstract earcons are particularly good for representing abstract notions in the user interface for which no intuitive mapping between sound and meaning exists. The presentation manager has an extensible set of predefined earcons which makes up a basic vocabulary and can alert the user to system state information and certain common messages from applications.

2.3.2 Auditory Icons [6]

Auditory Icons (representative earcons) are familiar real world sounds which have an intuitive mapping in the interface. The advantage of auditory icons is that with a proper mapping between sound and meaning, the learning aspect is eliminated. However, because real world sounds are being applied to a generally abstract computer environment, confusion can arise because inevitably some users might not be able to identify the sounds properly or could misunderstand the meaning of the sound in its context. The presentation manager has an extensible set of auditory icons which would be common in typical user environments.

2.3.3 Parameterized Music

Parameterized music is useful for keeping "watches" on certain interface parameters. The music is continuous, and the human ear is very sensitive to small changes in tempo, volume, and pitch [7]. The user is responsible for learning the mapping of variables to music parameters. The actual content of the music can be generated with some simple algorithms which can be tuned for user preference [8, 9].

2.3.4 Synthesized Speech

Speech is the most reliable auditory means to communicate very specific information. However, it requires a higher cognitive load than any of the previously mentioned auditory constructs. This can potentially be problematic in a busy auditory environment. The interaction of speech and nonspeech audio is examined in section 3.

2.3.5 Raw Audio Data

Sometimes the application programmer desires certain sound effects, or wishes to convey some form of information not supported by the presentation manager. In this case, a raw audio message can be sent, which, to the media selector, contains far less semantic information than a normal message. The representation of the message cannot be altered; the message will simply be passed on to the scheduler.

2.4 The Scheduler

The scheduler displays the auditory messages taking into account the *latency* and *urgency* parameters. If an audio message gets delayed for too long, the message is sent back to the media selector for re-representation since it is likely that the auditory state of the system has significantly changed and the original representation may no longer be a suitable choice. As messages age in the scheduler, their priorities are incremented slightly to avoid possible starvation.

3 Interaction of Different Sound Media

In order to display the various auditory media simultaneously in an effective way, the presentation manager must have some knowledge of how the different media interact with one another.

3.1 Historical Precedents

There are few precedents for this type of interaction in sounds generated by real world objects, so we have to look elsewhere to begin examining the problem. As has been done in other work in nonspeech audio [10, 11], we examined the historical precedents based on music to examine how related problems were solved. In this specific case, we examined how composers had intertwined voice and nonspeech audio. We specifically did not want to consider singing as voice, although there is a good deal to be learned from song that can be applied to the problem. There are a number of musical precedents for spoken dialogue with background music. It must be emphasized in our examples, that both the dialogue and the music were composed as one piece, rather than an alternation of voice and music as in Peter and the Wolf.

Our first example is from melodrama, a genre of musical theater found in 19th century opera. *Melodrama* is "a genre of musical theater that combined spoken dialogue with background music." [12] One of the most famous of these is by Carl Maria von Weber in the Finale of Der Freischutz. The intention of the finale was to create a diabolically eerie scene with the speaker, Caspar, evoking nature-pictures of frightening scenes. Diminished and augmented intervals as well as chromaticism are used in the melody and harmony. Another example we considered were two pieces written in the 20th century by Arnold Schoenberg, Ode to Napoleon and Survivor from Warsaw. We examined the interlacing of the voice of a narrator in both of these, but the music itself has elements of atonality or his 12-tone method, which we did not consider in our analysis. Lastly, we looked at an example from Rap music, Young M.C.'s Bust a Move.

3.2 The Analysis

The material can be analyzed from many aspects. There are two that are particularly interesting which we discuss briefly. The first is the semantic content of the dialogue and background and the second is the use of simultaneous or sequential voice and background sounds. At some points there are real-world sounds accompanying both voice and music. This occurs more often in Der Freischutz because it is an opera in which singers are acting out roles. Casper pounds and hammers as he works. Whips crack, horses neigh, and dogs bark in one of his images. The narrator in Survivor from Warsaw, recalls horrifying images as well, and in the background one can hear soldiers marching. Both of these pieces have choruses that suddenly begin during the narration. The mood of the voice and the music are reflected in one another.

Often when words are to be emphasized, the background sounds cease entirely and the voice is heard alone. Similarly, in Rap music, lack of a musical element functions to shift more emphasis to the spoken word. In Bust a Move, the bass guitar carries the melody line of the underlying music through much of the composition. However, toward the end of a chorus, the bass abruptly stops as the narrator speaks the emphasized phrase "bust a move" over the soft rhythm.

When the voice stops and music starts, a new idea may begin and the music is used to bridge the spoken sections. Music or real-world sounds often precede the voice as to introduce it. Short auditory messages or motifs often function to indicate a character's presence or recurring theme. If the voice is speaking nearly alone (sometimes soft, barely audible background sounds can be heard) as the emotional content increases, the music gets louder and definite themes are heard.

3.3 Application to Computer Interfaces

How can this be applied to the computer interface? We are doing some experimental work to examine the various effects mentioned above, that may be simple to apply in many different cases. The principles found in pieces such as the ones mentioned above can be carried over almost in their entirety. A theme which captures some image either through real-world sounds or with earcons can be heard before a voice is heard. This sets the scene for the message and alerts the listener to the spoken words. Sounds used during speech can emphasize important parts of the message or even impart information that is not spoken but that can be conveyed quickly through emotional response, such as the urgency of the message. Sounds structured using known musical constructions, such as hierarchical structure or transformations can quickly identify a series of related musical fragments to the listener [3]. On the other hand, perception of a missing nonspeech audio element can serve to emphasize the spoken word. Finally, sound that follows a message can be used to bridge messages and convey continuity between them. Real-world sounds can be used between messages to reinforce their content.

4 Example Application: Heart Rate Monitor

This program monitors the user's heart rate, and presents a constant auditory message which indicate that rate. Furthermore, it allows the user to set a "target" zone for which the heart rate is optimal for the current level of physical exertion, and also an upper bound for which it would be dangerous to allow the heart rate to exceed.

1. The *Heart Rate Indicator Message*
   This message indicates the current heart rate, and could be represented as a parameter in an algorithmically generated music background or as a tone whose volume or pitch is changed as the rate changes.

   - Perception Style: *Persistent*
   - Type: *Numeric Value*
     The heart rate (in beats per minute) is the value.
   - Urgency: *Medium-Low*
   - Latency: *100*
   - Precision: *Medium*

2. The *Target Zone Indicator*
   The message indicates that the user is in the specified rate zone. The message should be present whenever in the zone, and absent otherwise.

   - Perception Style: *Conditionally Persistent*
     The heart rate must be in the user specified zone for this message to be displayed.
   - Type: *Application State Indicator*
   - Urgency: *Medium-High*
   - Latency: *200*

- Precision: *Medium-High*

3. The *Below-Zone Event Marker*
   This message indicates that the user has just dropped below the target zone. Possible representations include an earcon, real world sound, or voice.

   - Perception Style: *Discrete*
   - Type: *Application Warning*
   - Urgency: *Medium-High*
   - Latency: *50*
   - Precision: *Medium*

4. The *Above-Zone Event Marker*
   This message indicates that the user has just dropped below the target zone. Possible representations include an earcon, real world sound, or voice.

   - Perception Style: *Discrete*
   - Type: *Application Warning*
   - Urgency: *Medium-High*
   - Latency: *50*
   - Precision: *Medium*

5. The *Over Maximum Threshold Event Marker*
   This message indicates that the user's heart rate has exceeded the users's maximum value. Harm will come to the user unless the heart rate is dropped immediately.

   - Perception Style: *Discrete*
   - Type: *Event Marker*
   - Urgency: *High*
   - Latency: *0*
   - Precision: *High*

5 Conclusion

Many of the problems introduced by allowing multiple applications to output audio can be solved with a centralized approach. By giving the applications programmer a higher level abstraction to encode information into sound, the burden of designing sound for the interface is lessened. Furthermore, the final result should be more perceptible to the user since the application programmer would not have been able to design the sonic output with any knowledge of the overall auditory context to which the new application was being added.

References

[1] Ephraim P. Glinert and Meera M. Blattner. Programming the multimodal interface. In *ACM Multimedia 93 Proceedings*, pages 189–206, Anaheim, CA, 1993. Association of Computing Machinery.

[2] Albert S. Bregman. *Auditory Scene Analysis*. MIT Press, Cambridge, MA, 1990.

[3] Meera M. Blattner, Denise A. Sumikawa, and Robert M. Greenberg. Earcons and icons: Their structure and common design principles. In *Human-Computer Interaction*, volume 4, pages 11–44, 1989.

[4] Stephen A Brewster, Peter C. Wright, and Alistair D. N. Edwards. A detailed investigation into the effectiveness of earcons. In Gregory Kramer, editor, *Auditory Display. Sonification, Audification, and Auditory Interfaces*, pages 471–498. Addison Wesley Publishing Company, 1994.

[5] Sheila M. Williams. Perceptual principles in sound grouping. In Gregory Kramer, editor, *Auditory Display. Sonification, Audification, and Auditory Interfaces*, pages 95–125. Addison Wesley Publishing Company, 1994.

[6] William W. Gaver. Auditory icons: Using sound in computer interfaces. *Human-Computer Interaction*, 2(2):167–177, 1986.

[7] Stephen Brewster. Providing a model for the use of sound in user interfaces. Technical Report 169, University of York, Heslington, York, Y01 5DD, June 1991.

[8] Peter S. Langston. Img/1: An incidental music generator. *Computer Music Journal*, 15(1), Spring 1991. Choose music style and length and the computer generates a peice in that style, with the arrangement differing by music style.

[9] Peter S. Langston. (201) 644-2332 or eedie & eddie on the wire, an experiment in music generation. In *Proceedings of the Usenix Summer '86 Conference*, 1986.

[10] Meera M. Blattner and Greenberg Robert M. Communicating and learning through non-speech audio. In Alistair D. N. Edwards and Simon Holland, editors, *Multimedia Interface Design in Education*, chapter 9, pages 133–143. Springer-Verlag, 1993. NATO ASI Series F.

[11] Meera M. Blattner, Robert M. Greenberg, and Minau Kamegi. Listening to turbulence: An example of scientific audiolization. In *Multimedia Interface Design*, pages 87–102. ACM Press/Addison-Wesley, 1993.

[12] Donald Jay Grout and Claude V. Palisca. *A History of Western Music*. W. W. Norton & Company, New York, fourth edition, 1988.

DATE
FILMED
10/5/94

END