Contextual Awareness, Messaging and Communication in Nomadic Audio Environments

Nitin Sawhney

M.S., Information Design and Technology
Georgia Institute of Technology, 1996

B.E., Industrial Engineering
Georgia Institute of Technology, 1993

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements of the degree of
Master of Science in Media Arts and Sciences
at the
Massachusetts Institute of Technology
June 1998

Author: ____

Program in Media Arts and Sciences
19 May 1998

Certified by: ____

Christopher M. Schmandt
Principal Research Scientist, MIT Media Laboratory

Accepted by: ____

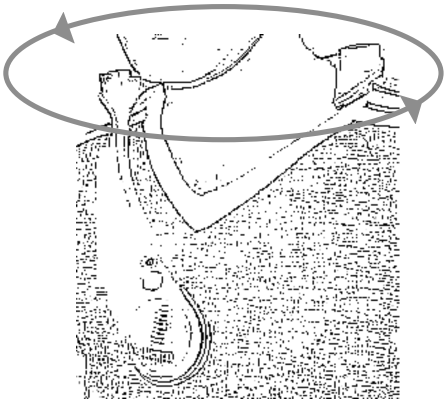
Stephen A. Benton
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Contextual Awareness, Messaging and Communication in Nomadic Audio Environments

Nitin Sawhney

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on May 19, 1998
in partial fulfillment of the requirements of the degree of
Master of Science in Media Arts and Sciences

ABSTRACT



Nomadic Radio provides an audio-only wearable interface to unify remote information services such as email, voice mail, hourly news broadcasts, and personal calendar events. These messages are automatically downloaded to a wearable device throughout the day and users can browse them using speech recognition and tactile input. To provide an unobtrusive interface for nomadic users, the audio/text information is presented using a combination of ambient and auditory cues, synthetic speech and spatialized audio.

A notification model developed in *Nomadic Radio* dynamically selects the relevant presentation level for incoming messages based on message priority, user activity and the level of conversation in the environment. Temporal actions of the user such as activating or ignoring messages while listening, reinforce or decay the presentation level over time and change the underlying notification model. Scaleable notification allows incoming messages to be dynamically presented as subtle ambient sounds, distinct *VoiceCues*, spoken summaries or spatialized audio streams foregrounded for the listener.

This thesis addresses techniques for peripheral awareness, spatial listening and contextual notification to manage the user's focus of attention on a *wearable audio computing* platform.

Thesis Supervisor: Christopher M. Schmandt
Title: Principal Research Scientist, MIT Media Laboratory

Thesis Committee

Thesis Supervisor:

Christopher M. Schmandt
Principal Research Scientist
MIT Media Laboratory

Thesis Reader:

Alex Pentland
Toshiba Professor of Media Arts and Sciences
Academic Head, MIT Media Laboratory

Thesis Reader:

Pattie Maes
Associate Professor of Media Arts and Sciences
MIT Media Laboratory

Thesis Reader:

Hiroshi Ishii
Associate Professor of Media Arts and Sciences
MIT Media Laboratory

Acknowledgments

Thanks to the following people for their support and contributions to this thesis:

Chris Schmandt, my advisor, for his insights and ideas about speech and audio interfaces. Chris displayed patience as I adjusted to the rigors of the academic environment at MIT and helped me develop the perseverance and commitment to do challenging work. He prescribed a rigorous and disciplined approach to research, which has proved very beneficial for me in the long term. Chris taught me the value of building real systems to solve problems in real environments.

Sandy Pentland has been a keen supporter of this work from the beginning. He provided the necessary encouragement and a strong academic motivation as I embarked on a new research area.

Pattie Maes served as a reader despite her time constraints. She took great care to provide critical and timely feedback. Taking her classes had a real impact on this work and my view of adaptive interfaces.

Hiroshi Ishii always advocated a rigorous yet imaginative approach to human interface design. Hiroshi took a personal interest in my work and his energy and enthusiasm have been a real inspiration.

Lisa Fast and **Andre Van Schyndel** at Nortel for supporting this project, by providing technical guidance, necessary hardware, and responsive feedback to make *wearable audio computing* a reality.

Remhi Post and **Yael Maguire** in the Physics and Media Group, and **Binh C. Truong** (a UROP student in our group) for help with *SoundBeam* modifications and 3D printing for the *Radio Vest*.

Zoey Zebedee from the Parsons School of Design, for collaborating on the *Radio Vest* and working late nights stitching together different configurations. **Linda Lowe** worked very hard to bring an excellent team of international fashion design students to work with researchers at the Media Lab in July-Oct. 1997.

The *wearables* community at the Media Lab, especially **Thad Starner**, **Brad Rhodes**, **Lenny Foner** and **Travell Perkins** for introducing me to wearable computing and brainstorming about its possibilities.

My fellow students in the *Speech Interface Group*: **Lisa Stifelman**, **Brenden Maher**, **Stefan Marti**, **Natalia Marmasse**, and **Keith Emnett** for providing an enriching environment and a supportive attitude.

Deirdre Fay for putting up with my endless requests for hardware, with a cheerful smile.

Rasil Ahuja (*Razz Dazz*) provided exhaustive feedback at late hours of the morning, reading and faxing each chapter from India. Her help made the writing much more coherent. **Laxmi Poruri** was meticulous in reading a draft of the whole thesis and provided precise comments to refine the overall content.

Tony Jebara, **Brian Clarkson**, and **Deb Roy** for endless brainstorming and giving me a big picture when I found myself lost in implementation details. I will always cherish their friendship. **Nuria Oliver**, **Marina Usmachi** and **Claudia Urrea** for being supportive friends in difficult times.

My parents, **Narinder** and **Pushpa**, and my sisters, **Pragati** and **Priti**, for their constant support, enthusiasm, and confidence in me throughout the years. And of course my roommate, **Avinash Sankhla**, for putting up with my crazy schedule and lifestyle at MIT.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

In an information rich environment, people access a multitude of content such as news, weather, stock reports, and other data from a variety of information sources. People increasingly communicate via services such as email, fax, and telephony. Such a growth in information and communication options is fundamentally changing the workplace and “beginning to have a seismic effect on people’s professional and personal lives” *(Pitney Bowes Study, April 8, 1997[1]).* A partial solution to information overload is to give people timely and filtered information, most relevant to the context of their current tasks [Maes94]. Seamless nomadic access to personal information and communication services [Schmandt94a] should be made available to users in a passive and unobtrusive manner based on their level of attention and interruptability.

1.1 Nomadic Access to Timely Information

Simple devices, such as pagers, provide a convenient form of alerting users to remote information. These devices offer an extremely low-bandwidth for communication, and subsequently the interface does not afford rich delivery of information content. Telephones are ubiquitous and cellular services offer mobility. Computer-based telephony services, such as *Phoneshell* [Schmandt93], offer subscribers integrated access to information such as email, voice mail, news and scheduling, using digitized audio and synthesized speech. However, telephones primarily operate on a synchronous model of communication, requiring availability of both parties. The user is charged while accessing services since a connection must be maintained while listening to the news or a stock report. All processing must be done on the telephony servers, rather than the phone itself. The requirement of synchronous connection prevents the device from continuously sensing its user and environment. Portable computing devices can download messages when not being actively used, allowing asynchronous interaction similar to email. Personal Digital Assistants (PDAs) offer the benefit of personal applications in a smaller size. However, they generally utilize pen-based graphical user interfaces which are not ideal when the user’s hands and eyes are busy.

Hand-held [Stifelman93, Roy96] and mobile [Stifelman96, Wilcox97] audio devices with localized computing and richer interaction mechanisms certainly point towards audio interfaces and networked applications for a new personal information platform, *Wearable Audio Computing* (WAC) [Roy97]. Wearable auditory interfaces can be used to enhance an environment with timely information and provide a sense of peripheral awareness [Mynatt98] of people and background events.

[1] *http://www.pitneybowes.com/pbi/whatsnew/releases/workers_overwhelmed.htm*

This thesis demonstrates interface techniques developed for wearable access and awareness of timely information in a nomadic physical environment. *Nomadic Radio* is a wearable computing platform that provides a unified audio-only interface to remote services and messages such as email, voice mail, hourly news broadcasts, and personal calendar events. These messages are automatically downloaded to the device throughout the day and users can browse through them using voice commands and tactile input. To provide an unobtrusive interface for nomadic users, the audio/text information is presented using a combination of ambient and auditory cues, synthetic speech and spatialized audio.

The goal of this thesis is to develop the appropriate interface techniques for wearable interaction as well as an infrastructure for unified messaging on the wearable. The key issue addressed is that of handling interruptions to the listener in a manner that reduces disruption, while providing timely notifications for contextually relevant messages.

1.2 Why use Wearable Computing for Contextual Messaging?

Wearable computers can be considered a form of “smart clothing” if they provide comfortable, usable and unobtrusive access to computing. Several characteristics of wearable devices, suggested by researchers and active wearable users [Starner97a][Rhodes97], make them ideal for capture and delivery of personal information relative to devices such as pagers, PDAs, phones, and portable computers. These characteristics include:

- *Unobtrusive Usage in Everyday Environments*
- *Active Information Access and Passive Awareness*
- *Environmental and Affective Sensing*
- *Adaptive Interaction based on Continuous Long-term Usage*

Wearable computers are generally always operational and can monitor the user and the environment on a continual basis. Unlike PDAs, they do not need to be turned on or woken up each time a new task must be performed. Hence, interaction with wearables is direct and performed on a continuous basis. Interfaces such as head-mounted displays, speech I/O and auditory feedback emphasize hands-free operation. Users can request information on-demand or simply be made aware of new messages or events. Sensors such as GPS or IR (infrared) receivers can provide positioning information while affective sensors on the user's body can determine the state of the user. Long term interaction with such devices provides an opportunity to personalize information and the interface, as the system learns and adapts to the user’s daily routine.

1.3 Why use Speech and Audio in Nomadic Environments?

Most wearable computers today derive their interfaces from concepts in desktop computing such as keyboards, pointing devices, and graphical user interfaces. If wearable computers are expected to become as natural as clothing, we must re-examine the interface with different standards of usability. In this thesis, we will consider the role of speech and audio as the primary interface modality for wearable access to unified messaging and timely notifications.

1.3.1 Scalability

Traditional input/output modalities such as keyboards and screens lose their utility as devices get smaller. The functionality and ease of use of GUIs does not scale well on small, wearable devices. Hence, new I/O modalities must be explored to provide natural and direct interaction with wearable computing. Speech and audio allow the physical interface to be scaled down in size, requiring only the use of strategically placed speakers and microphones [Stifleman93] and perhaps tactile input for privacy and in noisy environments. Yet, scalability is also an issue with an audio modality where listeners may wish to hear varying levels of information content in different situations. In chapter 5, we will discuss techniques for scaleable auditory notification of text and audio messages.

1.3.2 Unobtrusive Operation

Consider the head mounted display which is used as the primary output device of most wearable computers. One criticism is that such a display is far too noticeable and therefore socially unacceptable to be considered a serious solution (although commercial systems are quickly driving down the size of such devices). A more serious objection regards the type of perceptual load the display places on the user. There will be situations in which the user's eyes are busy, although she is otherwise able to attend to information from her wearable computer, such as when walking or driving. An “eyes-free” approach, using audio-based augmentation of the physical environment, can be used to express information directly. Spoken commands provide a natural and hands-free operation. This allows the user to simultaneously perform other tasks while listening or speaking [Martin89]. Such an interface can be unobtrusive and perhaps even designed to be discreet. Speech recognition is used in *Nomadic Radio* and we will consider design solutions for two different wearable configurations.

1.3.3 Expressive and Efficient Interaction

Voice is more expressive and efficient than text because it places less cognitive demands on the speaker and permits more attention to be devoted to the content of the message [Chalfonte91]. The intonation in human voice also provides many implicit hints about the message content. *VoiceCues* (short audio signatures), played as notifications in *Nomadic Radio,* indicate the identity of the sender of an email message in a quick and unobtrusive manner.

Speech provides a natural and direct means for capturing user input. It is faster to speak than to write or type. Momentary thoughts can be easily recorded before they are forgotten. Wearable devices with digital audio output and speech input can provide efficient forms of interaction for several user tasks. For example, in a conversation or lecture, the audio transcript can be recorded transparently rather than using a slow text-entry device which might distract the user. Audio retrieval techniques can be used to access useful information at a later time.

1.3.4 Peripheral Awareness

People using wearable devices must primarily attend to events in their environment, yet need to be notified of background processes or messages. People have a good sense of awareness of background events and can easily shift their focal attention to significant events. Speech and music in the background and peripheral auditory cues can provide an awareness of messages or signify events, without requiring one’s full attention or disrupting their foreground activity. Audio easily fades into the background, but users are alerted when it changes [Cohen94]. In *Nomadic Radio*, ambient auditory cues are used to convey events and changes in background activity.

1.3.5 Simultaneous Listening

One can attend to multiple background processes via the auditory channel as long as the sounds representing each process are distinguishable. This well known cognitive phenomenon, known as the "Cocktail Party Effect" [Handel89] provides justification that humans can monitor several audio streams simultaneously, selectively focusing on any one and placing the rest in the background. A good model of the head-related transfer functions (HRTF) permits effective localization and externalization of sound sources [Wenzel92]. Yet the cognitive load of listening to simultaneous channels increases with the number of channels. Experiments show that increasing the number of channels beyond three causes a degradation in comprehension [Stifelman94]. Bregman claims that stream segregation is better when frequency separation is greater between sound streams [Bregman90]. Arons suggests that the effect of spatialization can be improved by allowing listeners to easily switch between channels and pull an audio stream into focus, as well as by allowing sufficient time to fully fuse the audio streams [Arons92]. The use of such spatial audio techniques for browsing and *scanning* personal messages will be demonstrated in *Nomadic Radio*.

1.3.6 Limitations of Speech and Audio

Speech input is a natural means of interaction, yet the user interface must be carefully devised to permit recognition and recording on a wearable device. Issues related to privacy and level of noise in the environment constrain speech input as well as audio output on wearables. In these situations tactile interaction can be used to control the interface and messages *foregrounded* (discussed in section 4.7.6). Speech is fast for the author but slow and tedious for the listener. Speech is sequential and exists only temporally; the ear cannot browse around a set of recordings the way the eye can scan a screen of text and images. Hence, techniques such as interactive skimming [Arons93], non-linear access and indexing [Stifleman93, Wilcox97] and audio spatialization [Mullins96] must be considered for browsing audio.

Design for wearable audio computing requires (1) attention to the affordances and constraints of speech and audio in the interface (2) coupled with the physical form of the wearable itself. The physical design and social affordances of the wearable audio interface play an important role in determining how it will be adopted in specific situations or environments and its usage over a long term and continuous basis.

1.4 Research Challenges

For this thesis there are three key research challenges (described in chapter 3) briefly discussed here:

1.4.1 Unified Access to Heterogeneous Information Services

Nomadic users want remote access to a number of disparate information sources that they traditionally use on their desktop. Personal information such as email, voice mail and calendar as well as broadcast services such as hourly news, traffic reports and weather information, should be seamlessly delivered on a wearable system. Designing an interface that permits users to browse all such messages easily is essential. In *Nomadic Radio*, a modeless interface permits all operations on messages of any category and dynamic views allow the message space to be restructured, based on the user's interest.

1.4.2 Unobtrusive and Peripheral Wearable Audio Interface

To provide rich information services to users on the run, an appropriate non-visual interface is important. A well designed audio interface provides peripheral awareness of notifications and presentation of messages, using a combination of auditory cues, synthetic speech and simultaneous audio techniques. Integrating such techniques in a coherent manner and synchronizing them with voice and tactile-based control, is a key challenge for a wearable audio interface.

1.4.3 Knowing When and How to Interrupt the Listener

Any user wearing an active audio device at all times would be continuously disrupted by incoming messages and audio broadcasts. Hence, filtering and prioritization techniques must determine timely information for the user. In addition, the system should infer the availability of the user to listen to such timely information, based on her recent activity and the context of the environment. An adaptive notification model is needed to dynamically present relevant information at the right time.

1.5 Overview of Document

An overview of all chapters and an explanation of the typographical and symbolic conventions used in the document is provided below.

1.5.1 Summary of Chapters

Chapter 2, "Related Work," describes relevant work in telephone-based messaging, mobile audio devices, use of contextual information in wearable computing and recent augmented audio reality systems.

Chapter 3, "*Nomadic Radio* Design Overview," discusses the key aspects of the system regarding its audio interface, dynamic views for unifying text and audio messages, wearable audio design, and briefly introduces techniques for managing user interruption. Finally, an extensive demonstration of *Nomadic Radio* gives an understanding of its general functionality and interface techniques.

Chapter 4, "*Nomadic Radio* Interface Design," describes the main design criteria and techniques for modeless interaction. Detailed descriptions are provided for design of interface techniques based on speech recognition, tactile input, speech feedback, auditory cues and spatial audio. In addition, we will consider an auxiliary visual interface to complement messaging in the audio-only modality.

Chapter 5, "Scaleable and Contextual Notification," considers problems with interruptions and notifications with current communication technologies. A scaleable approach is proposed for presentation of messages, based on sensing the user and the environmental context. Dynamic operational modes allow the wearable system to reduce notifications and conserve resources. Finally, we will discuss techniques for reinforcement of notifications based on user actions and show an evaluation of the model.

Chapter 6, "System Architecture," describes the client-server interaction that allows the system to provide robust and flexible messaging. The different servers used in *Nomadic Radio* are discussed along with a strategy for using local and distributed interface services.

Chapter 7, "Conclusions," discusses the lessons learned and the key contributions of the work in *Nomadic Radio*. The chapter shows how some of the techniques can also be utilized for browsing and notification in everyday communication devices. Finally, we will consider future work towards location awareness, communication, and environmental audio classification.

1.5.2 Typographical Conventions

This document includes many transcripts of spoken interaction between *Nomadic Radio* and the user. These transcripts are indented and shaded for easy reading. Annotations are placed in bracketed text.

> **Nitin:** ***"move back" <or presses the back key>***
>
> **NR: <audio cue + VoiceCue>** ***"Unread short personal message 30 from Tony Jebara..."***

Figure 1.1: Formatting of spoken transcripts in *Nomadic Radio,* with annotations.

Actual text messages and system operational feedback is shown in bold Courier font.

```
 Priority: 0.266667 Activity: 0.143104 Speech Energy: 0
 Notify Level: 0.46992 Mode: audio cues - Threshold:0.41629
```

Figure 1.2: Formatting of operational feedback from *Nomadic Radio.*

Additional symbols used for auditory presentation in *Nomadic Radio* include the following:

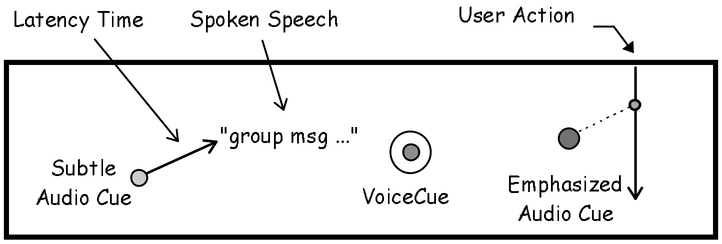


Figure 1.3: Conventions used for symbols representing varying levels of auditory cues, VoiceCues, synthetic speech and actions of the user when a message arrives.

2. Related Work

This chapter surveys prior research and commercial projects for telephone-based messaging, mobile audio access, wearable computing as well as recent work in augmented audio. We consider open research issues which help to situate the development of the proposed wearable audio platform, *Nomadic Radio*.

2.1 Telephone-based Messaging Systems

The existing telephony infrastructure provides nomadic users with ubiquitous access to information and communication services. Several telephone-based systems provide advanced speech and audio interfaces to such services. Recently introduced cellular devices offer enhanced interfaces and services.

2.1.1 Voiced Mail

Text and voice messages were first integrated for phone-based access in the *Voiced Mail* system [Schmandt84]. Messages were sorted by sender and played to the user, independent of the medium. A graphical interface added at a later time [Schmandt85] provided a unified messaging system accessible on a desktop as well as over the phone. A key aspect of the system was easy interruptability by the user during speech-only interaction.

2.1.2 Phoneshell

A nomadic computing environment was developed at the Speech Interface Group to provide remote access to a set of desktop applications via telephony. *Phoneshell* [Schmandt93] allows subscribers to access email, voice mail, calendar, rolodex and a variety of news, weather and traffic reports. Interaction is provided via touch-tones (DTMF). Users hear messages and feedback via synthetic speech and audio playback. *Phoneshell* is heavily used by members of the group on a daily basis, hence reliability and immediate access are important to subscribers. Despite the breadth of applications and nomadic access, *Phoneshell* is constrained by the limitations of the telephone keypad. However, once key combinations were learned, experts can execute actions easily using robust DTMF recognition. *Nomadic Radio* utilizes much of the infrastructure developed for *Phoneshell*, complementing the remote phone-based access of *Phoneshell* with spatial audio display and voice-controlled interaction for wearable inter-office access.

2.1.3 Chatter

Chatter [Ly94] is an attempt to extend the functionality of *Phoneshell* using continuous, speaker-independent speech recognition. *Chatter* focuses on communication within a workgroup, i.e. reading email, sending voice messages to workgroup members, looking up people in a rolodex, and placing outgoing calls. A discourse model tracks the user's place in a dialogue and preserves context among separate tasks. One of *Chatter's* major contributions is the incorporation of memory-based reasoning

(MBR) techniques. *Chatter* makes predictions about the user's next action based on observing the user's prior history. By reducing much of the interaction to "yes" and "no" responses, recognition errors were reduced. However, since *Chatter* does not verify the output of the recognizer, it is prone to erroneous actions such as deleting messages by mistake or unpredictably hanging up on the user during a session.

2.1.4 SpeechActs

SpeechActs [Yankelovich94] combines the conversational style of *Chatter* with the broad functionality of *Phoneshell*, providing a speech interface to email, calendar, stock quotes and weather forecasts. It allows users to access messages by number ("read message 13") and verifies irreversible requests such as "delete message". It provides progressively more detailed assistance if the system failed to understand the user. *SpeechActs* offers message filtering and categorization, allowing the user to skip over or delete uninteresting groups of messages.

2.1.5 WildFire

The *WildFire*[2] electronic assistant is a commercial system (1995) which offers functionality similar to prior projects in the Speech Interface Group. *WildFire* screens and routes incoming calls, schedules reminders and retrieves voice mail (but not email). It allows users to sort messages, yet the navigation is sequential. The discrete-word speech recognition in *WildFire* limits conversational style for interaction.

2.1.6 MailCall

MailCall [Marx96a] is a bold attempt towards effective telephone-based conversational messaging. The system provides remote access to email and voice messages, using speech recognition and a combination of intelligent message categorization, efficient presentation, and random access navigation. A key feature of the system, CLUES [Marx 95,96b] allows prioritization of incoming email messages based on the user's current interests as inferred from correlation with entries in the calendar, rolodex and recent email replies. To support random access of messages, *MailCall* dynamically updates the speech recognizers vocabulary. Inevitable recognition errors are handled by an interface algorithm which verifies potential mistakes and allows the user to correct them easily. *MailCall* also provides customized feedback to users based on the conversational context of the discourse. *Nomadic Radio* utilizes CLUES to prioritize email messages.

2.1.7 AT&T PocketNet

PocketNet[3] is a wireless information access service introduced by AT&T (in late 1997), that allows users to browse their email and subscribed information on an enhanced phone with text display, buttons for navigation and a CDPD radio modem. It provides a personal organizer with calendar and quick dialing from a rolodex, synchronized with the user's desktop. Online information sources provide news, travel, sports and financial information downloaded to the device periodically.

[2] *http://www.wildfire.com*

[3] *http://www.attws.com/nohost/data/pocketnet/pn.html*

2.1.8 Nokia GSM Phones

The Nokia 9000 *Communicator*[4], a digital cellular phone (introduced in Sep. 1997), enables users to send and receive faxes, e-mails, short messages, Internet services and access corporate and public databases. Other applications include an electronic calendar, address book, notepad and calculator. The *Communicator* is a specialized phone running the GEOS operating system and requires GSM-based service. Interaction with these applications requires a built-in keyboard and small display (see figure 2.1).

Nokia recently introduced the *6100 series*[5] GSM phones (Nov. 1997) with a large graphics display, advanced call management, and one touch voice mail. A novel feature of these phones is the *Profile* function which enables users to adjust the phone settings according to various situations, with caller grouping and caller group identification through different ringing tones and graphics (see figure 2.1). For example, by selecting the "Meeting mode", the phone alerts only for priority calls, while "Outdoors mode" alerts louder and can be set to only allow private calls through. The user can define different ringing tones for various caller groups, such as friends and colleagues. The calendar in the Nokia 6100 phones allows the user to record appointments or write notes.

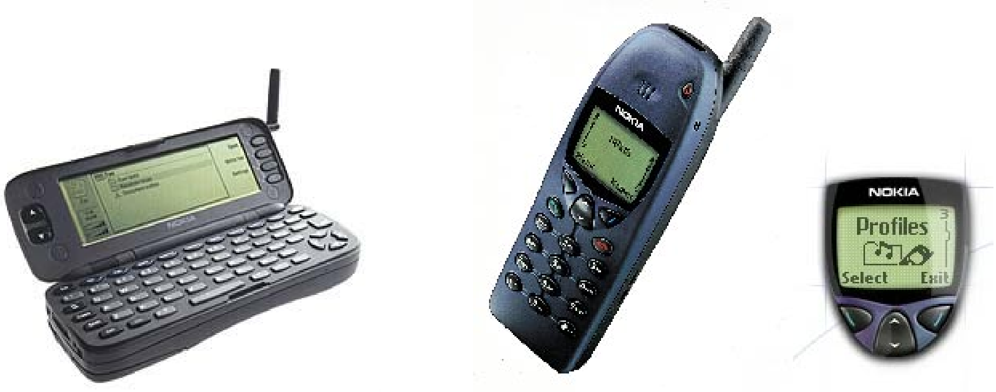


Figure 2.1: Nokia’s digital cellular phones: the 9000 Communicator, with built-in messaging and personal organizer, and the 6100 phone with Profiles and Caller Groups.

Profiles and Caller Groups on such phones allow the user some control over notifications. Yet most users will be less inclined to continuously create caller groups and set modes for profiles. In *Nomadic Radio*, a contextual notification model automatically scales the level of spoken feedback and auditory cues for incoming messages, based on the user’s activity and level of conversation in the environment. The user can set priorities for preferred people (but is not required to do so) or the priority of a message can be inferred by the system using intelligent filtering.

[4] *http://www.nokia.com/com9000/n9000.html*

[5] *http://www.nokia.com/6100launch/home.html*

2.1.9 General Magic Portico

Portico[6] is a phone and web-based service from General Magic (being introduced in mid-1998), that enables users to access, retrieve and redistribute information across computer and telephone networks. *Portico* integrates voice mail, email, address books, and calendars as well as Internet-based information such as news and stock quotes. This information is automatically filtered and accessed through a natural language speech interface called *magicTalk*. The speech interface uses conversational dialogues and a dynamically extensible grammar. General Magic is working with manufacturers of Windows CE and other hand-held devices to bundle elements of the *Portico* service with their products.

To summarize, such telephone-based messaging systems generally require asynchronous usage, i.e. users must dial-in to the messaging service each time they wish to browse their messages. Hence, there is less incentive to use the system on an on-demand basis or for short transactions. Specialized phones from AT&T and Nokia act as personal organizers, but do not provide an unobtrusive hands-free interface, intelligent filtering of information or contextual notification. In contrast, a wearable approach (discussed in section 2.3) provides a system that is always listening to the user, minimizing any initiation time to access messages. It provides on-demand notification of timely messages. By sensing the user and environmental context at all times, a wearable messaging system can decide when it is most appropriate to interrupt the listener and scale its feedback accordingly.

2.2 Mobile Audio Devices

Mobile devices like PDAs have limited audio functionality, however some recent prototypes of mobile applications allow users to capture and access audio recordings synchronized with their handwritten notes on a portable device. Hand-held audio devices allow voice recordings to serve as memory aids and also deliver structured and personalized audio programs. Several efforts have recently been initiated to provide information and communication services to drivers in automobiles.

2.2.1 FiloChat

FiloChat [Whittaker94] indexes speech recording to pen-strokes of handwritten notes taken with a digital notebook. Users can access a portion of audio data by gesturing at an associated note.

2.2.2 Audio Notebook

The *Audio Notebook* [Stifelman96] is a paper-based augmented audio system (see figure 2.2) that allows a user to capture and access an audio recording synchronized with handwritten notes and page turns. The *Audio Notebook* also structures the audio via topic-changes and phrase-breaks, which is particularly helpful for browsing audio from classroom lectures.

[6] *Poticio was previously code-named "Serengeti" by General Magic. http://www.genmagic.com/portico/portico.html*

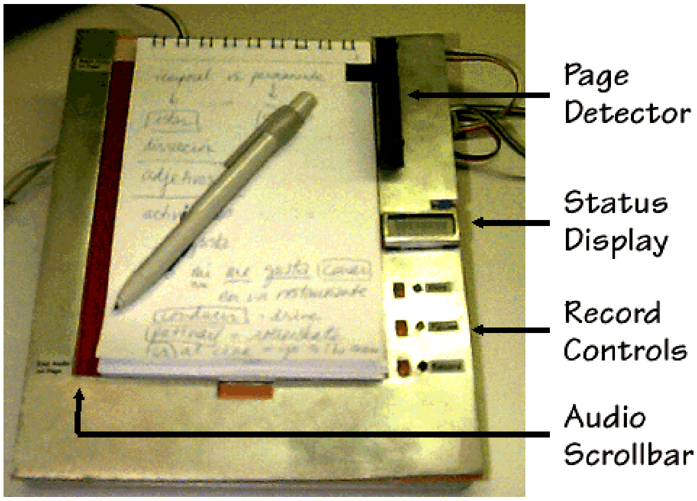


Figure 2.2: A prototype of the Audio Notebook, that allows a user to capture, skim, and browse an audio recording in conjunction with notes written on paper.

2.2.3 Dynomite

Dynomite [Wilcox97] is a portable electronic notebook which captures, searches and organizes handwritten and digital audio notes. *Dynomite* utilizes user-defined ink properties and keywords to dynamically generate new structured views of the recorded information. Synchronized audio is selectively captured and edited via user's highlighting of the audio for minimal storage on mobile devices.

2.2.4 VoiceNotes

VoiceNotes [Stifelman92, 93] was designed as an application for a voice-controlled hand-held computer which allows creation, management and retrieval of user-authored voice notes. The user interacts with the system using a combination of button and speech input, and feedback is provided via recorded speech and non-speech audio cues. In *VoiceNotes*, recorded notes are organized as lists under user-defined categories. The user's spoken category name became a voice command to allow random access to notes. *VoiceNotes* is a key influence in the interface design of *Nomadic Radio*. Techniques developed for recorded voice notes such as modeless navigation, scanning lists and speech feedback, are enhanced in *Nomadic Radio* for effective interaction with unified text and audio messaging.

2.2.5 NewsComm

NewsComm [Roy 95, 96] delivers personalized news and audio programs to mobile listeners through a hand-held playback device. Audio servers receive audio programs form various audio sources and processes them to extract structural information. Structural descriptions of the news programs are generated by locating speaker changes and analyzing pause structure. The server selects a subset of available recordings to download into the local memory of the hand-held device (based on user preferences and listening history) when the user connects the device to the server. The interface consists of button-based controls (see figure 2.3) to allow the user to select recordings, indicate which ones were interesting and navigate within a recording using structural information. *Nomadic Radio* assumes a wireless connection to provide continuous downloads of the user's subscribed services such as email, voice messages, news and calendar events, thus automatically synchronizing the desktop and nomadic information views. *Nomadic Radio* does not utilize structured audio for navigation, but techniques such as *spatial scanning* and *foregrounding* allow simultaneous playback and scaleable presentation of audio and text messages. Instead of utilizing listener profiles, messages once loaded in *Nomadic Radio* are dynamically presented based on a contextual notification model. While *NewsComm* utilizes a robust button-based interface for navigation, *Nomadic Radio* provides a hybrid interface that permits use of both speech recognition and/or buttons.

2.2.6 Audio Highway's Listen Up Player

The *Listen Up Player*[7] from Audio Highway (1997), is a hand-held digital recorder and player that delivers personalized audio content (see figure 2.3). The company's web site allows consumers to choose from audio-only selections that can be downloaded to a PC and stored for later mobile playback. Unlike *NewsComm*, no structured descriptions of the audio programs are generated to aid navigation. Audio content is advertising-supported and therefore delivered free-of-charge to consumers. (For every hour of selected audio content, consumers receive three minutes of audio advertising messages - six 30-second spots.). Unlike *NewsComm*, the *Listen Up Player* does not capture any listener for effective selection of programming.

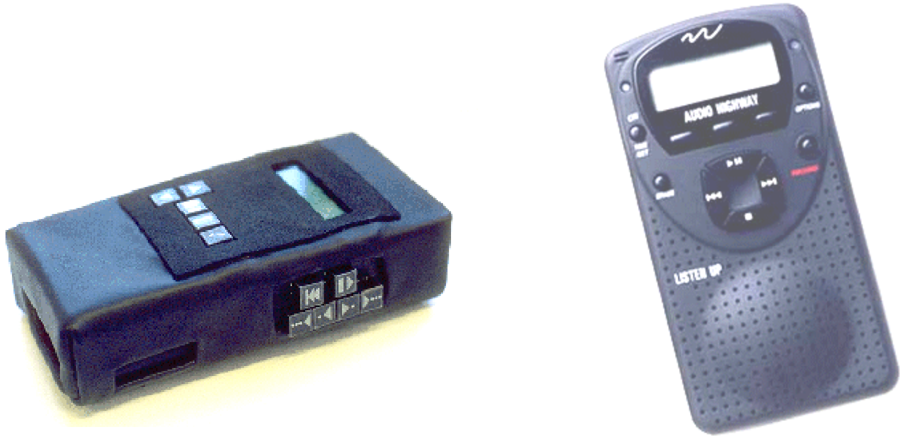


Figure 2.3: The NewsComm hand-held device developed at the MIT Media Lab (1995) and the Listen Up Player from Audio Highway (1997).

[7] *http://www.audiohighway.com/*

2.2.7 In-Vehicle Communication and Navigation Systems (Telematics)

Currently there are several efforts underway to provide wireless information services to drivers on the road, using mobile audio interfaces in automobiles. The Clarion *AutoPC* is a Windows CE-based system integrating communication, navigation, information and entertainment for the automobile. Clarion, in collaboration with Microsoft, has proposed a system (to be released in Q3, 1998) to provide real-time news updates and traffic information using a wireless network. Text-to-speech is used to provide status information, email and assistance to the driver. The interface is controlled via buttons on the faceplate (see figure 2.4) and speech recognition to provide simple functions such as making a call from the rolodex. An optional navigation system will provide direction guidance using GPS-based position sensing.

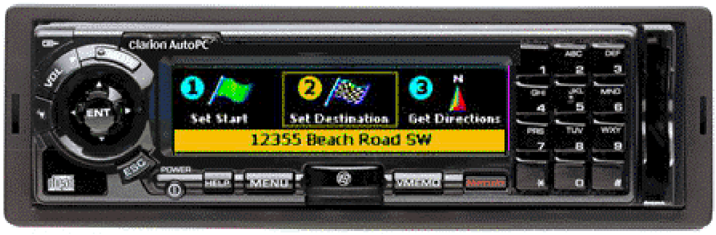


Figure 2.4: The Clarion *AutoPC*, a mobile communication, navigation and information system for automobiles. The faceplate provides buttons and controls for interaction, and a LCD screen with an icon based user-interface. Voice commands will allow hands-free control, along with text-to-speech for hearing status information.

In February 1998, Motorola established the Telematics Information Systems (TIS), a new business organization responsible for leveraging the company's leading technologies related to the emerging *Telematics* market. *Telematics* is considered a new way of using wireless voice and data to provide drivers and their passengers with location-specific security, information, and entertainment services from a central service center. Motorola announced a partnership with Nissan Motor Corporation to introduce the *Infiniti Communicator*, an in-vehicle communications system (IVCS) available on their luxury sedans in March 1998. The system merges wireless communication and global-positioning satellite technologies to offer drivers 24-hour emergency and roadside assistance, air-bag deployment notification, stolen-vehicle notification and remote door unlocking via the Infiniti Response Center (IRC).

Currently there are efforts underway by the TIS group to provide personal messaging and information services using voice recognition, text-to-speech and audible indicators. One of their key concerns[8] is how to measure and minimize distraction to the primary driving task, from information interaction and notifications. A key design criteria for *Nomadic Radio* is the notion of handling user interruption based on her focus of attention inferred from activity, conversations and priority level of information. Such a contextual notification model can be generalized for use in mobile devices and *Telematics* interfaces.

[8] *Personal discussions with Bob Denaro and Cheuck Chan from Motorola's TIS Group, during a recent visit to the MIT Media Lab (April 22, 1998)*

2.3 Wearable Computing and Audio Augmented Reality

Nomadic users want continuous access to relevant information using natural and unobtrusive interfaces. We consider visual approaches for providing timely information and recorded memories on wearables based on the user's context. We will then focus on audio-only wearable interfaces that augment the physical environment with active audio information as well as peripheral auditory cues.

2.3.1 Remembrance Agent

The *Remembrance Agent* (RA) [Rhodes97] is a program that continuously analyzes the user's typed responses on a wearable (via a chording keyboard) to query the user's database of email and textual notes. The RA displays one-line summaries of notes-files, old email, papers, and other text information that might be relevant to the user's current context. These summaries are listed in the bottom few lines of a heads-up display, so the wearer can read the information with a quick glance. By using a relevance measure based on the frequency of similar words in the query and reference documents, the RA provides timely and sometimes serendipitous information to the wearable user. Hence, the RA shows the benefits of a wearable context-driven augmented reality.

2.3.2 Augmented Hyper-links and Audio/Video Memories in DyPERS

In one wearable approach at the MIT Media Lab [Starner97a], visual tags in the wearer's physical environment identified objects and rendered the associated text, graphics or video content on the user's visual field. Such physical hypertext links used location context to provide instructions on use, history or information left by previous users. A recent approach called *DyPERS* [Jebara98], utilizes vision techniques to provide audio/video memories of a user's experience on a museum visit. A key innovation is the use of real-time image recognition techniques to trigger the user's own recordings related to physical objects in the environment.

In a recent paper [Starner97a], researchers suggest the use of sensors and user modeling to allow wearables to infer when users would not wish to be disturbed by incoming messages. However, the system should understand enough information context to alert the user of emergency messages immediately. The authors refer to work by Chris Schmandt related to identifying time-critical messages. They suggest an approach based on waiting for a break in the conversation to post a summary of an urgent message onto the user's heads-up display.

In *Nomadic Radio*, a purely non-visual approach is utilized to present timely information to listeners, based on the acoustic context of the environment[9]. Messages are filtered and prioritized, summaries conveyed as spoken text, and auditory cues are presented based on the inferred level of user interruption.

[9] *The notion of using conversational level to defer interruption by messages was first suggested by Brad Rhodes and Thad Starner in personal discussions with them regarding audio-only wearable computing (November 1996). Deb Roy later suggested using environmental audio as a means for establishing the general context of the user's location.*

2.3.3 DIE AUDIO GRUPPE: Performances with Electro-acoustic Clothes

Since 1983, Benoît Maubrey and his Berlin-based *AUDIO GRUPPE*[10] have been building electro-acoustic clothing and suits. These are clothes equipped with loudspeakers, amplifiers, and 257 K samplers (see figure 2.5) that enable them to react directly with their environment. In addition, they wear radio receivers, contact microphones, light sensors and electronic looping devices in order to produce, mix, and multiply their own sounds and compose these as an environmental concert. *Audio Jackets* (1982) used second hand clothing onto which loudspeakers had been sewn. These first prototypes are equipped with portable cassette players and 10-watt amplifiers, playing pre-recorded cassettes. *Audio Ballerinas* (1989-1996) were created using solar powered digital samplers to allow performers to record live sounds around them. Sensors and receivers integrated on the Plexiglas surfaces of the tutus allowed the *Audio Ballerinas* to create an entire spectrum of sounds via their clothing. The performers use rechargeable batteries and/or solar cells which ensures them complete mobility both indoors and outdoors.

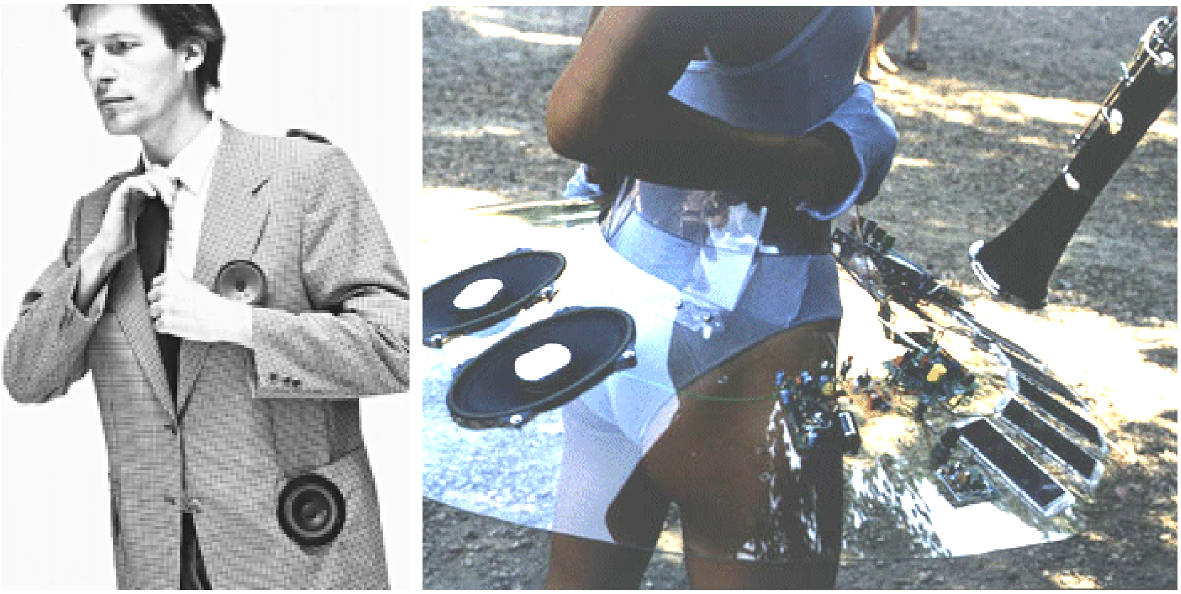


Figure 2.5: Benoît Maubrey's *Audio Jackets* (1982) and *Audio Ballerinas* (1989-1996) used electronic sensors and loudspeakers to record and create electro-acoustic performances with their clothes.

Benoît's performances boldly explored the unique affordances of a variety of wearable audio configurations, and represent the earliest form of interactive audio wearables[11]. Benoît's work served as an inspiration for the *Radio Vest* designed for *Nomadic Radio*.

[10] *http://www.inx.de/~maubrey/*

[11] *Personal discussions with Benoît Maubrey at the International Symposium of Electronic Arts (ISEA) in Chicago, September 1997.*

2.3.4 Ubiquitous Talker

Ubiquitous Talker [Rekimoto95] is a camera-enabled system developed at Sony Research Labs. It provides the user information related to a recognized physical object via a display and synthesized voice. The system also accepted queries through speech input.

2.3.5 Augmented Audio Museum Guide

A prototype audio augmented reality-based tour guide [Bederson96] presented digital audio recordings indexed by the spatial location of visitors in a museum. This is a early implementation of a wearable audio system which stores only pre-defined audio information to augment a physical environment. This system did not utilize the listener's context or prior listening patterns to selectively filter or present information.

2.3.6 SpeechWear

SpeechWear [Rudnicky96] is a mobile speech system developed at CMU which enabled users to perform data entry and retrieval using an interface based on speech recognition and synthesis. A speech-enabled web browser allows users to access local and remote documents through a wireless link.

2.3.7 Audio Aura

An augmented audio reality project at Xerox PARC, *Audio Aura* [Mynatt98], explored the use of background auditory cues to provide serendipitous information coupled with people's physical actions in the workplace. The project leveraged an existing infrastructure of active badges and distributed IR sensors along with wireless headphones to deliver audio cues to people in the workplace. The long-term goals of the system included use of multiple information sources (such as calendar and email) and multiple means for triggering the delivery of auditory information. The services in *Audio Aura* were designed to be easy to author, customize and lightweight to run. Hence, *Audio Aura* represents a light-weight version of [Bederson96], where local audio storage is not required and a user's location is transmitted via the active badge infrastructure. The design of sound environments and techniques were explored in a simulated aural environment (using VRML). To provide audio in the periphery, alarm sounds were eliminated and a number of harmonically coherent "sonic ecologies" were explored, mapping events to auditory, musical or voice-based feedback. Such techniques convey events, such as the number of email messages received, identity of senders or groups, and abstract representations of group activity as a continuous backdrop.

In contrast to these approaches, *Nomadic Radio* is developed as a wearable audio platform (local and distributed) with high-bandwidth audio interface services which allow active navigation of unified messages, as well as peripheral auditory notifications. The use of speech recognition, spatial audio, synthetic speech and ambient and auditory cues provide rich forms of interaction and scaleable presentation. Continuous sensing of the user and environment as well as filtering and prioritization of incoming information, allows the system to utilize a contextual notification model. Actions of the user reinforce the model and adjust its notifications over time, based on the user's preferred interruption level in specific contexts.

3. Nomadic Radio Design Overview

This chapter presents an overview of *Nomadic Radio* and a conceptual design of the messaging and notification framework. First we consider how a variety of information services are integrated in *Nomadic Radio*. The interface must provide a unified means for navigating these services and delivering notifications in a nomadic environment. A description of the wearable platform will give an understanding of the physical form and affordances of such a device. Results from an evaluation study by Nortel will be discussed, to compare earphones, headsets and alternative solutions. Undesirable interruption is a key problem that must be addressed to allow a user to readily adopt such wearable devices. An approach based on situational awareness and contextual notification, will be briefly introduced here (it is discussed in more detail in chapter 6). Finally, a demonstration will illustrate the functionality of *Nomadic Radio* for messaging and notification as well as the design of interface for use in a non-visual and wearable device.

3.1 Creating a Wearable Audio Interface

Nomadic Radio provides a unified audio interface to a number of remote information services. Messages such as email, voice mail, hourly news broadcasts, and personal calendar events are automatically downloaded to the device throughout the day. The user can select a category and browse the messages within it. To provide a hands-free and unobtrusive interface to a nomadic user, the system primarily operates as a wearable audio-only device (see figure 3.1). Here information and feedback is provided to the user through a combination of auditory cues, spatial audio playback and synthetic speech.

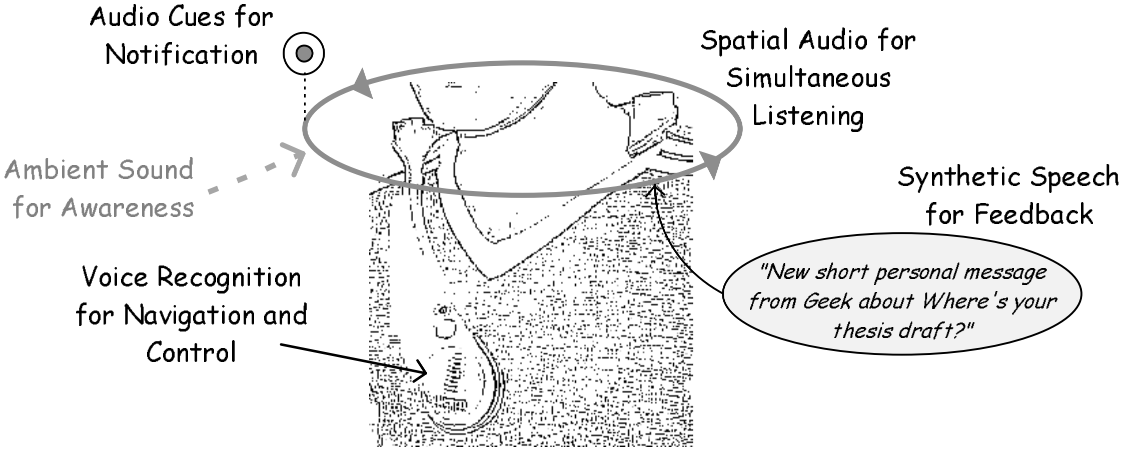
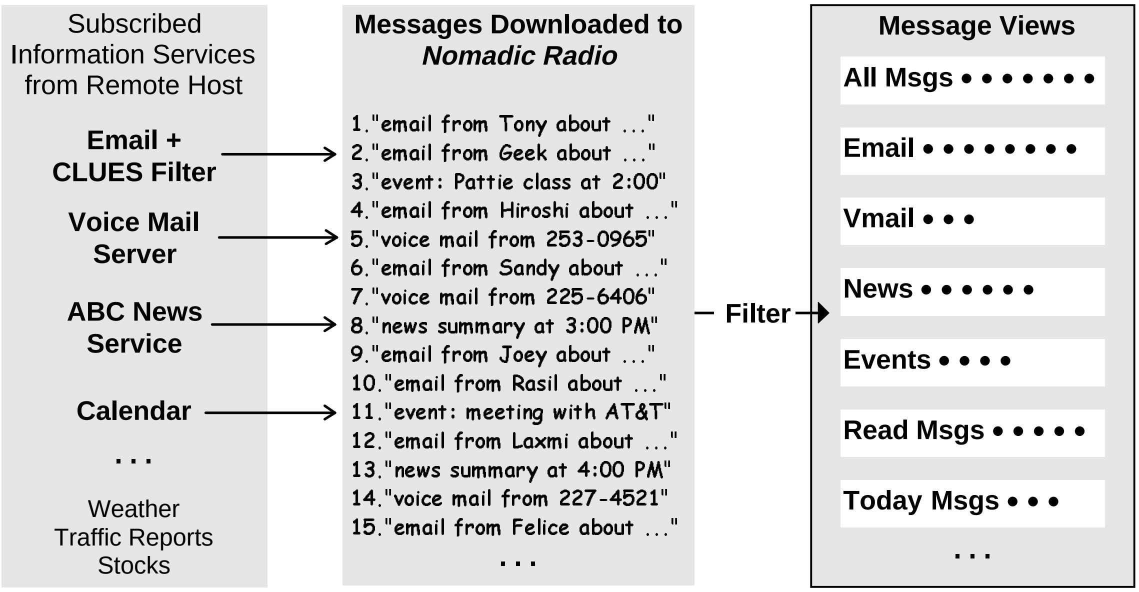

Audio Cues for Notification
Spatial Audio for Simultaneous Listening
Ambient Sound for Awareness
Synthetic Speech for Feedback
Voice Recognition for Navigation and Control
"New short personal message from Geek about Where's your thesis draft?"

Figure 3.1: All messages are browsed in a unified manner via speech, audio cues and spatial audio. The interface is controlled via voice commands and tactile input.

Textual messages, such as email and calendar events are spoken in a concise manner. Special emphasis has been placed on the design of appropriate auditory cues to indicate system activity, message notification, confirmations and break-downs. Auditory cues becomes especially important when synthetic speech feedback must be scaled back if the user is inferred to be busy. Figure 3.1 illustrates some of the interface techniques and modalities used in *Nomadic Radio* for delivering dynamic notifications to the user and allowing her to browse a variety of messages. In *Nomadic Radio*, audio sources such as voice messages and hourly news broadcasts are played within the user's listening environment. Several such broadcasts can be presented simultaneously as spatialized audio streams, to enable the listener to better segregate and browse multiple information sources. Users can navigate messages and control the interface using voice commands, coupled with tactile input in noisy environments or social situations.

3.2 Unifying Information Services through Dynamic Views

A unified messaging interface permits all operations to be performed on any category of messages. Messages are dynamically structured within categories by filtering and creating views based on attributes such as message type, unread status, priority or time of arrival. Figure 3.2 shows a conceptual overview of different views of information sources generated by such filtering. Such a framework offers a modular approach towards structuring new information services added in the future. It provides a flexible means for generating new views using different message attributes and allows recursive filtering of existing views. For example, the user may say "go to my voice mail," request "view today's messages" and then say "view unread messages". This dynamically filters and restructures the existing messages into new hierarchical subsets of the overall messages.



1."email from Tony about ..."
2."email from Geek about ..."
3."event: Pattie class at 2:00"
4."email from Hiroshi about ..."
5."voice mail from 253-0965"
6."email from Sandy about ..."
7."voice mail from 225-6406"
8."news summary at 3:00 PM"
9."email from Joey about ..."
10."email from Rasil about ..."
11."event: meeting with AT&T"
12."email from Laxmi about ..."
13."news summary at 4:00 PM"
14."voice mail from 227-4521"
15."email from Felice about ..."
. . .

Figure 3.2: Messages dynamically structured within information categories or filtered views based on message attributes and user interests.

3.3 Designing a Wearable Audio Platform

Audio output on wearable computers requires use of speakers worn as headphones or appropriately placed on the listener’s body. Headphones are not entirely suitable in urban environments where users need to hear other sound sources such as traffic or in offices where their use is considered anti-social when people communicate face-to-face. Earphones (worn on a single ear) are discreet, but do not allow effective delivery of spatial and simultaneous audio. In these situations, speakers worn on the body can instead provide directional sound to the user (without covering the ear). However, they must be designed to be easily worn and least audible to others. For *Nomadic Radio*, several solutions were considered before adopting a design based on research at Nortel. We will first discuss results from their design evaluations.

3.3.1 Nortel’s Evaluation of Personal Listening Devices

Nortel conducted an evaluation[12] [Nortel94] of listening devices such as headsets (corded and cordless), speaker phones, cordless phones, and single-ear earphones (the *Jabra*). These were compared with a variety of solutions for hands-free devices worn around the neck, based on what would come to be called the *SoundBeam Neckset*. A series of user discussion groups were conducted with participants (business users and call center agents) to understand the perceived differences between the devices and to determine the specific design attributes that deliver perceived audio privacy, audio quality, comfort, and image. The evaluation results revealed several key issues:

- Telephone handsets were the least preferred of all devices due to their lack of mobility and an inability to work and talk. In addition, privacy and perceived "dead-time" for callers were issues.
- Users expressed concerns regarding the *Jabra* earphone; irritation or infection from over a long period, the cord breaking easily or getting caught. The earphones also provoked a negative image of using a Walkman or wearing a hearing aid. However the small size was a factor in a preference for the *Jabra*’s discreet appearance and usage.
- Due to their weight, headsets are associated with causing headaches and are tiresome to wear all day. Headsets tend to interfere with women’s hair and earrings. The image of looking like a receptionist was considered negative and especially inappropriate while speaking with others.
- Wearability as defined by perceived comfort, sturdiness and image was attributed by users to two specific design attributes: (1) the perceived distance of the front of the device from the neck, and (2) device integration, i.e. if it looks like one piece (if the device curves around the neck it should end somewhere along the arc of the curve). Wires were seen as inhibiting wearability and should be avoided whenever possible. "The smaller the better" was also true for wearable solutions.

[12] *Proprietary Nortel Report TL940041, "SoundBeam Design Qualification" (Sept., 1994). Excerpted here with written permission from Nortel. Please contact Lisa Fast <lfast@nortel.ca> for further details.*

- The *Neckset* design raised some doubts among some users about privacy and whether the microphone would pickup ambient noise in addition to the speaker's voice. Some users considered the "high-tech" look somewhat disturbing and felt the device might interfere with clothing and accessories. In addition, long hair rubbing against the microphone and speakers may be a problem.
- The *Neckset* provided many benefits over other devices. One aspect was easy wearability, i.e. being able to put it on and take it off easily. Its one-piece plastic design gave the perception of robust components, hence the *Neckset* seemed easier and more durable to handle. As a new solution, it was not associated with any preconceived stereotypes by most users, although the "high-tech" look was considered an asset by some.

Most users valued hands-free and cordless usage and showed low tolerance for headsets and ear pieces worn on a long term basis. Users preferred privacy of conversation and good audio quality. Range of usage (being able to move form office to office) and control over the interface (dialing, volume control and mute button) were also important criteria. Freedom of movement (without breaking) and aesthetically pleasing look of the design were important for most users.

Overall, the design of the *SoundBeam Neckset* appeared very acceptable to users and seemed beneficial relative to solutions available at the time. Packaging and marketing strategies were considered important to convey the audio quality and privacy aspects of the *Neckset* (directional speakers do in fact provide private listening) and also to allow users to overcome potential concerns about the perceived image while wearing the *Neckset.* However, such concerns would be reduced over time as the *Neckset* is more readily worn by early adopters in an inter-office environment. In Nortel's design evaluation, the *Necksets* gained popularity after they were actually tried on by the participants, as the solutions were found to be "more comfortable than they looked".

3.3.2 Adoption of the Nortel SoundBeam in Nomadic Radio

The *SoundBeam Neckset* is the primary audio I/O device used in *Nomadic Radio*. The suite of *SoundBeam* technologies are protected by patents filed by Nortel. The *Neckset*, a research prototype embodying the first implementation of the *SoundBeam* technology, was originally developed for use in hands-free telephony and desktop communications. It consists of two directional speakers mounted on the user's shoulders, and a directional microphone placed on the chest (see figure 3.3). Nortel allowed us to modify the *Neckset* and evaluate it in the context of wearable audio computing.

The original *Neckset* was adapted[13] for use in *Nomadic Radio* by patching the amplifier circuit to allow two independent audio channels for stereo output. Direct audio I/O from the wearable PC was added to

[13] *We wish to thank Lisa Fast, Andre Van Schyndel and Richard Levesque at Nortel for providing hardware and circuit diagrams for the SoundBeam Neckset and CAD files for the speaker enclosures. Thanks to Remhi Post and Yael Maguire in the Physics and Media Group, and Binh C. Truong (a UROP in our group) for help with SoundBeam modifications and 3D printing for the Radio Vest.*

deliver spatialized audio to the listener and allow voice recognition on the *Neckset*. Spatialized audio is rendered in real-time by *Nomadic Radio* and the left-right stereo channels are delivered to directional speakers on the *Neckset*. A button on the *Neckset* could be used to activate speech recognition or deactivate it in noisy environments. Currently this functionality is provided via wireless tactile input or using voice commands to put the recognition in "sleep" mode.

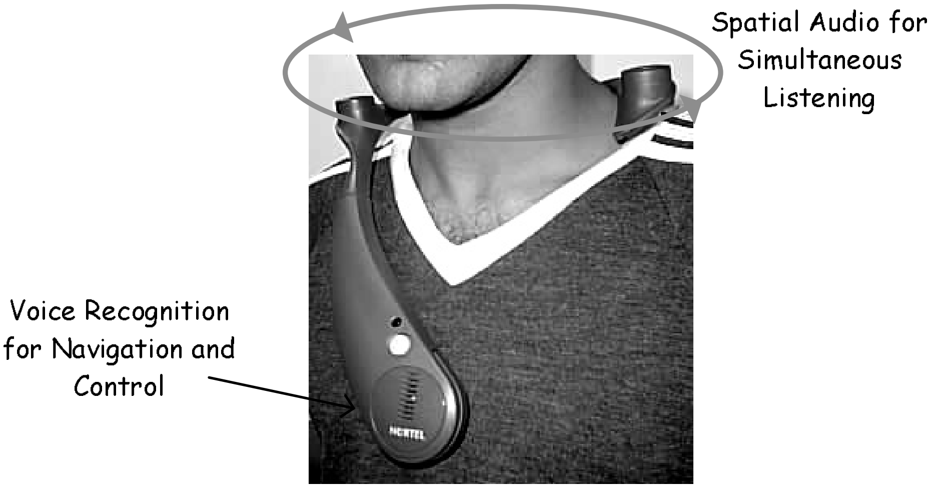


Figure 3.3: The Nortel *SoundBeam Neckset*, the primary wearable audio device adopted in *Nomadic Radio* for inter-office use. The directional speakers are utilized for rendering spatialized audio to the listener, and a directional microphone allows speech recognition.

In October 1997, we incorporated elements of Nortel's *Neckset* into a new solution, based on a collaboration with Zoey Zebedee, a fashion designer from the Parsons School of Design, New York. The speaker enclosures were molded on a 3D printer to provide better directional audio. The audio components were integrated into a new wearable and modular configuration called the *Radio Vest,* designed for a more rugged and mobile usage (see figure 3.4). Here the clip-on speakers and microphone modules can be easily detached when not needed. This configuration also delivers sufficient quality for spatialized audio and the sound enclosures ensure private listening for the user.

The *Radio Vest* is an experimental prototype for outdoor wearable use, whereas the *SoundBeam Neckset* remains the primary device for inter-office use of *Nomadic Radio*. We must evaluate the ergonomics and social affordances of such designs because they are worn more readily around the Media Lab, and considered appropriate refinements. For instance, what social conventions must be used to address the wearable in public spaces? How can others be made aware of the user listening to and hence distracted by, an incoming message especially when it is inaudible to them (a flashing light on the wearable)? By wearing and using *Nomadic Radio* on a regular basis in social environments, many such issues can be understood better.

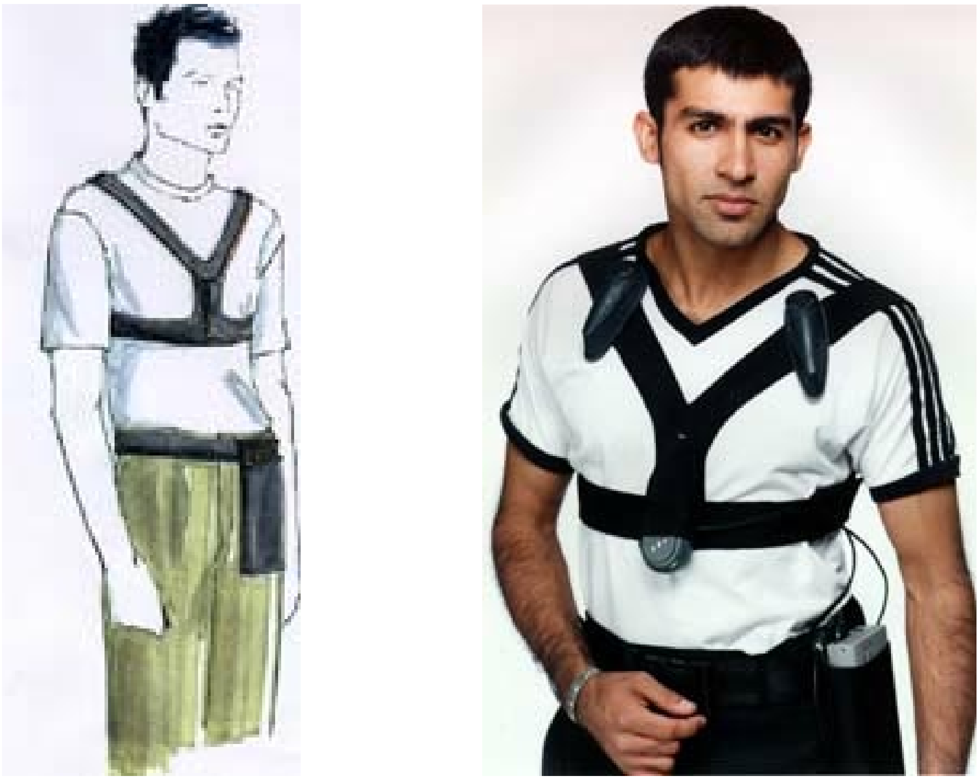


Figure 3.4: The *Radio Vest*, conceptual design and final implementation of an outdoor wearable configuration with clip-on directional speakers and powered microphone.

A long-term evaluation and ethnographic study in the future is warranted. However, in the near-term one can consider the use of personal audio devices such as cordless phones and pagers, and their effect on a user's lifestyle and social interactions. Such devices provide many benefits but also pose a number of problems (see sections 5.1 and 5.2) regarding undesirable interruption that inhibit their effective use. These issues are discussed in detail in chapter 5, and are briefly introduced below.

3.4 Knowing When and How to Interrupt the Listener

If a user is expected to wear an audio device that continuously informs her of incoming messages and news broadcasts, these will sometimes be rather disruptive. Most users don't tolerate cell-phones and pagers that go off in public or without any warning. Hence, these are usually turned off and thereby become ineffective for timely messaging. In *Nomadic Radio*, the notification approach is based on the user's inferred attentiveness and availability. Contextual notification is based on three key aspects, i.e. the priority of the information, level of recent usage of the device, and the conversational level in the environment (discussed in section 5.6.3). These factors allow the system to dynamically scale the notification and decide when a message should be conveyed to the listener.

The user can tune the responsiveness of the notification model using an interactive visual graph or simply by setting pre-defined interruption levels (high, medium or low). In addition, when the user hears a notification and finds it undesirable, her actions provide negative and positive reinforcement to dynamically change the underlying notification model. A wearable device has limited power, processing and memory resources which must be conserved if it is to be used effectively on a continuous basis. *Nomadic Radio* operates under a number of dynamically changing modes which gradually transition the device to reduced states of operation and notification. Chapter 5 describes the scaleable presentation and contextual cues for the notification model as well as the dynamic operational states in *Nomadic Radio*.

3.5 Listening and Speaking on the Run: A Demonstration

A demonstration of the current version of *Nomadic Radio* is shown here to give the reader a better understanding of the functionality and interface design discussed in chapters that follow. This design has been refined after several iterations based on periodic usage by the author and informal feedback from others during demonstrations.

3.5.1 Initiating Nomadic Radio the First Time

When *Nomadic Radio* is launched, it tries to initiate network connections with local and remote servers for services such as synthetic speech, speech recognition, audio monitoring and the wireless keypad interface. Once *Nomadic Radio* connects with the local speech module, it provides synthetic speech feedback (see figure 3.5). It notifies the user about the status of remote servers that were not reachable or files not found on the server. It then retrieves the user's personal messages based on services she has subscribed to, such as email, voice messages, and calendar events. While loading, the user hears the ambient sound of water punctuated with long and short splashes indicating messages retrieved. After it has finished loading, it waits momentarily before playing a summary of the user's last message.

Wednesday, 10:07 AM

Nomadic Radio (NR): <audio cue indicating it was launched successfully>

NR: ***"Cool, I can listen and talk now."***

NR: <audio cues for successful connections to remote services> + ***"Could not connect to audio classifier!"***

NR: ***"Nomadic Radio launched. Loading your messages, Nitin. Please wait ..."*** **<ambient sound of flowing water with splashes while downloading messages>**

NR: ***"Loading your calendar entries ..."***

NR: ***"Nitin, you have 8 unread messages out of 32 total messages and 3 scheduled events today."***

NR: <pauses a few seconds> <audio cue for group message> + *"Last group message from Walter Bender about guest speaker at NIF Lunch today."*

Figure 3.5: When *Nomadic Radio* is first launched it notifies the user of available services and the status of messages being loaded via audio cues and synthetic speech.

3.5.2 Waking-up Nomadic Radio and Interacting Spontaneously

The user starts speaking with a colleague for a few minutes. If the user stops using the system for some time, it turns off speech synthesis and recognition and goes into sleep mode, however it can monitor the user and be activated when spoken to. The scenario below (figure 3.6) demonstrates the user actively navigating categories and browsing messages using voice commands. Tactile input is provided when speech recognition is less reliable in noisy environments. Summaries of messages are spoken and distinct auditory cues indicate their urgency, as well as whether voice commands were actually recognized.

Wednesday, 1:15 PM

Nitin says: *"Nomadic Wake Up!"*

NR speaks: *"Ok, I'm Listening."*

Nitin: *"Go to my messages"*

NR: <audio cue for command understood> *"Nitin, you have 17 unread messages out of 40 total messages and 3 scheduled events today."* <waits momentarily> <audio cue for a most important message followed by the related VoiceCue> *"Last very important message 40 from Sandy Pentland about reply to my thesis draft".*

Nitin: *"move back" <or presses the back key>*

NR: <audio cue + VoiceCue> *"Unread short personal message 30 from Tony Jebara about Lets hit the gym?"*

Nitin: *"Read this message"*

NR: <audio cue> *"Message Preview: Tony Jebara, says I'm heading for the gym in 15 minutes, where are you?"*

Nitin: *"Go to my calendar"*

NR: <audio cue> *"Nitin, you have 3 scheduled events today."* <pauses momentarily> *"Special Event 3: Meeting with AT&T at 2:00 PM for 30 minutes".*

Nitin: *"Nomadic Sleep!"*

NR: <audio cue> *"Ok, I'll stop listening now."*

Figure 3.6: A user wakes up *Nomadic Radio* and actively browses messages and calendar events by navigating to different categories.

3.5.3 Notifying the User of Incoming Messages

The user goes to the meeting at 2:00 PM. When messages arrive, the system scales down all notifications if the user has been using the system for a while or if the level of conversation in the room is high. Group messages are not heard and personal messages are notified as audio cues, but timely messages may be played if the conversation level goes down slightly. If the user is engrossed in the meeting when the message plays, a discreet key press stops playback (see figure 3.7). The system infers that it provided an undesirable interruption when the user was busy, so it turns down notifications for future messages.

Wednesday, 2:17 PM

NR: <audio cue for timely message> <sound of water heard flowing faster> ***"New short timely message from ...."***

Nitin: <presses the stop key before the system even finishes speaking the summary>

NR: <audio cue indicating playback stopped>

Figure 3.7: User interrupts a message summary being spoken during the meeting. The system then turns down all future notifications to avoid interrupting the user.

The user goes back to the office and starts reading a paper. Moments later, an important voice message arrives. While loading, the user hears the ambient sound of water playing faster, indicating that its a long audio file (see figure 3.8).

Wednesday, 2:43 PM

NR: <ambient sound of water flowing faster for a few seconds + audio cue "phone ringing" + waits momentarily> ***"New voice message from 253-0352"*** **<plays a 2.5 second preview of the audio, before fading it away>** ***"Hi Nitin, this is Felice ..."***

Nitin: <ignores message while he finishes reading> <moments later he responds> ***"Play Preview" <or presses the preview key>***

NR: <audio cue "phone ringing"> ***"Hi Nitin, this is Felice. The meeting with AT&T got moved to the Garden conference room. Its still at 2:00 PM though. Bye."***

Figure 3.8: Nomadic Radio notifies the user about a voice message and plays a short preview. The user later activates the message to hear it in its entirety.

3.5.4 Scanning News Summaries

At certain times of day the user likes to hear news summaries. Updated audio news summaries from ABC Radio are downloaded to the device seven minutes past every hour. However, a summary is played only if the user is not busy and it is the preferred listening time. Voice messages, hourly news broadcasts and *VoiceCues* for email are played in a specific spatial direction around the listener’s head, based on the time of message arrival (this technique is described in the section on spatial listening). The user can browse previous news broadcasts or simply scan all broadcasts to quickly hear about the day's major news stories. Each news summary is heard fading in from a distinct spatial position, and played for a few seconds before fading out, while the next one fades in (see figure 3.9). This graceful listening effect continues until all messages are played or the user deactivates scanning (either by reading a message or pressing stop).

Wednesday, 3:07 PM

NR: <ambient sound of water flowing faster for 30 seconds>

NR: <audio cue> <The sound of news is heard fading-in from the right side of the user's listening space and the familiar station identifier is played .... *"This is ABC News, I'm Tim O'Donald. President Clinton said in his news conference today ...."* <audio gradually fades away after 4/5th of the message is heard>

Nitin: "Scan messages"

NR: <audio cue> *"Message scanning activated"* <plays each news summary from the current one to the first received in that day> *"This is ABC News, I'm Bob Heart. Independent prosecutor Ken Starr said today ..."* <the message plays and gradually fades away> *"... for the ABC information Network, I'm Bob Heart."* <the next news summary is heard in an appropriate spatial location, overlapping momentarily with the previous one> *"This is ABC News, I'm Tim O'Donald ... "*

NR: <moments later a voice message arrives> <ambient sound of water flowing faster for a few seconds + audio cue "phone ringing" + waits momentarily> *"New voice message from Geek 253-5156"* <plays a 2.5 second preview of the audio, before fading it away> *"Hey this is Chris ..."*

Nitin: *"Play Foreground"*

NR: <audio cue "phone ringing" + voice message played in foreground while the news fades into the background> *"Hey this is Chris, our server beacon seems to be having some problems, can you look into it?"*

Figure 3.9: User hears a News summary and scans previous ones (each playing in a distinct spatial location). When an incoming voice message arrives, it is heard in the foreground while the news fades to the background.

3.5.5 Cleaning-up Messages and Terminating

The user can delete any message while browsing or say "clean-up my messages" to delete all previously read and group messages. Every 24 hours, the system also checks for all the read and group messages, automatically deleting them for the user. Later, these messages can be removed from the server. *Nomadic*

Radio continues running throughout the day and automatically switches to varying levels of sleep to conserve power and reduce notifications. Hence, the user rarely needs to power down the system unless the machine needs to be rebooted for some reason. At that point, all unsaved messages are saved and deleted messages removed from the server. The system disconnects from all remote servers and gracefully terminates its operation (see figure 3.10). During termination the user can abort at any time and keep the system running.

Wednesday, 11:47 PM

Nitin: *"Clean-up my messages."*

NR: *"I'm cleaning up your messages, Nitin ... "* <audio cue for cleaning up> *"Deleted 14 messages, 9 were read and 5 were group messages."*

Nitin: *"Terminate Nomadic Radio"*

NR: *"Ok, I'm quitting now ..."* <pauses> *"Removing 14 messages from server"* <pauses> *"Saving 36 messages on server"* <pauses> *"Disconnected from remote speech server"* <pauses> *"Terminating Nomadic Radio. Nitin, Good Bye!"*

Figure 3.10: User cleans up all messages (also done automatically every 24 hours) and terminates *Nomadic Radio,* which quits gracefully with spoken feedback.

These scenarios demonstrate the key features of the audio interface in providing appropriate notifications and confirmations via auditory cues and speech feedback. The user interacts with the system using a combination of voice commands and key presses. The general pace of interaction is carefully designed to allow the user to hear all audio cues and spoken feedback. Simultaneous audio streams are gracefully managed to move the incoming one to the foreground as others are gradually faded back. There is sufficient time to respond to incoming messages. In the next chapter, we will examine the overall design criteria and these audio interface techniques in greater detail.

4. Nomadic Radio Interface Design

This chapter presents the key interaction techniques for controlling the interface and managing the user's listening environment in *Nomadic Radio*. A variety of auditory techniques is supported in a non-visual wearable interface. Synthetic speech enables explicit feedback in conjunction with speech recognition that provides hands-free navigation and control. Tactile input provides discreet control of the interface in situations where privacy or social interruption is an issue, as well as in noisy environments where speech recognition is poor. In addition to speech feedback, auditory cues provide effective awareness and notifications. Spatial listening is used for browsing and scanning messages easily via dynamic *foregrounding* and *scanning* techniques. These auditory techniques are designed to function synchronously without overwhelming or confusing the listener.

Finally, an auxiliary visual interface was developed to augment the audio-only interaction in *Nomadic Radio* for managing its operational characteristics. Visualization of messages over time provides an overview of the temporal message space and allows users to interactively tune the contextual notification model (discussed in chapter 6). We will first discuss the design criteria for using such techniques in the context of a wearable audio system and focus on a *modeless* strategy for all interaction in *Nomadic Radio*.

4.1 Design Criteria

Some general design criteria are used to inform the overall design of interaction techniques in *Nomadic Radio*. These criteria are based on usage of the wearable device by nomadic listeners who require instant and on-demand use of the device in an unobtrusive manner.

4.1.1 Always On, Listening and Easily Woken-Up

Wearable computers are designed to be operational at all times, sensing the user and the environment and delivering timely information. Since they are worn by the user, it is assumed that the wearable can be used at any time with minimal start-up or transition time. For the interface to be natural and responsive, the speech recognizer must always be listening to the user, unless the user explicitly requests the system to stop listening (in a noisy environment). *Nomadic Radio* operates under several dynamically changing operational modes to conserve resources (since it must always be on). The system should be woken-up easily by the user or when it needs to convey timely notifications.

4.1.2 Nomadic and Unobtrusive Interaction

Users of *Nomadic Radio* will primarily operate the device in situations where they are away from their desktop and simply need brief information via quick transactions. This means that the system must allow input and presentation of information using techniques requiring minimal interaction by the user. The user should be able to use the interface using a variety of modalities such as speech-only, when her hands and eyes are busy, or via tactile interaction or when speech input in the environment is not feasible.

4.1.3 Graceful Feedback and Presentation

Any system that is continuously worn must know when and how to interrupt the listener, to avoid annoying her when she is busy in a conversation or otherwise occupied. We will consider techniques for contextual notification and scaleable feedback. While the user is navigating the interface, the feedback must be concise yet natural to hear [Hayes83]. The synchronization, foregrounding and backgrounding of audio cues, speech feedback and auditory streams is critical for providing a coherent and pleasing presentation. We will consider timing issues and spatial audio techniques to allow effective browsing and previewing.

4.1.4 Provides Peripheral Awareness

Users of a wearable system are usually engaged in other tasks which require greater focus of attention. Hence, the wearable should be able to provide subtle notifications and feedback to minimize distractions to the listener. Background awareness of events can be provided using ambient auditory cues to indicate notifications and changes in activity of the system.

4.1.5 Consistent Interface for Scaleable Content

One goal of *Nomadic Radio* is to provide a number of personal messaging and information services. Any interface for accessing these services should be consistent for ease of use and to avoid confusion. This is especially beneficial as users switch between information and as new services are added in the future.

4.1.6 Navigable and Interruptible

In a non-visual interface, speech and audio will generally be slow and sequential. Hence, the user must be allowed to navigate the content by easily browsing between categories and messages. In addition, the user must have full control to stop audio playback or spoken speech at any time.

4.1.7 Situational Awareness

A wearable system is constantly on and hence can monitor the context of the user and the environment. Filtering information and correlating it with important aspects of the user’s context allows the system to gauge the urgency of potential notifications. Contextual cues inferred in this manner provide a powerful means of making the interface more responsive and adaptive to the user over time.

We will now consider the design of interface techniques based on these design criteria.

4.2 Modeless Interaction and Navigation

Users browse their messages by first selecting a *view* (category) that the message would be included in and then navigating sequentially through them. All messages can be accessed within the default *messages* view. Other views include *email*, *voice mail*, *news* or *calendar*. The messages in these views can be further filtered based on user queries such as *unread messages* or *today’s messages* to generate new reduced views of the message space. *Nomadic Radio* provides a *modeless interface* where all actions are valid at any point in time, for both text and audio messages. Commands such as "move {forward | back}", "play {summary | preview | full message}", and "{stop | play | speed-up } audio" are valid in any message context. Figure 4.1 shows a list of all commands available to the user in the system. The functionality of these individual commands are explained in greater detail in the sections that follow.

Command	Definition
"Go to my { view }"	**View is the filtered set of messages based on messages such as *Email*, *Voice Mail*, *News*, and *Calendar* events. Additional sub-views include *Unread* and *Today’s* messages as well as *priority*-based views.**
"Move { direction }"	**Allows the user to move sequentially *Forward* or *Back* through the messages in a view.**
"{ action } Message"	**Provides overall message browsing commands such as *First*, *Last*, *Scan,* and *Clean-up*. In addition, users can act on an individual message by saying *Read* or *Remove this message.***
"Play { how }"	**Plays a message as a *Summary*, *Preview* or the *Full message.* The user can also play the message in the *Background* or *Foreground* (changes spatial position for an audio message or volume of synthetic speech for a text message).**
"{ control } Audio"	**The audio navigation commands include *Play, Stop, Loop, Pause, Resume, Skip forward, Skip backward, Speed-up, Slow-down, Reset* (speed). These commands apply to both audio and spoken text messages.**
"{ control } Speech"	**The overall speed and volume of synthetic speech feedback can be set independently of any message by commands such as *Louder*, *Softer*, *Faster*, *Slower*.**
"Nomadic { listen switch }"	**The system actively starts listening to the user when it hears *Nomadic Wake-up* and stops listening when it hears *Nomadic***

	Sleep. (both are infrequently used phrases).
"{ talk switch } Talking Now"	*Start talking now* and *Stop talking now* allows the system to activate or deactivate speech feedback.
"Set { level } Interruption"	Set interruption level for notifications to *High*, *Medium* or *Low*. This changes all the weights for the notification model.
"Toggle { setting }"	Various environmental settings can be toggled such as *Ambient mode, Audio cues, Speech feedback* In addition, the system can be explicitly put to *Sleep* mode. The *Clock Display* can be toggled from 12 to 24-hours and the messages are dynamically re-spatialized accordingly.
"{ remote command } from Server"	Server process running on remote host computers can be controlled by using commands such as *Load | Save | Remove Messages or Download ABC News* or *Download My Calendar from server.* These functions are usually performed by the system automatically, but the user is also allowed to do so.
"I am confused, what can I say?"	This allows the user to request the overall commands that can be spoken to the system at any time.
"Help { command }"	Provides spoken instructions on using specific commands, described in the high-level help.
"Terminate Nomadic Radio"	This permits the user to exit the application. The system disconnects from all servers and saves/removes all messages and saves the action data before exiting.

Figure 4.1: Modeless command definitions in *Nomadic Radio*. These commands are available at all times via voice recognition and on-screen buttons. A somewhat smaller subset of the commands is available via tactile input from a wireless keypad.

These commands can be accessed via voice recognition (see section 4.3) and on-screen buttons for use in debugging the system and when speech recognition is not operational (see section 4.8). A limited subset of these commands have also been mapped to a wireless keyboard (see section 4.4) for use in situations where speech is less feasible such as high noise environments or lectures/meetings where user privacy and disruption to others is a concern. An iterative design process allowed selection of relatively intuitive commands that could be easily learned and especially utilized in a modeless manner.

In contrast, *moded interfaces* provide only a subset of all actions at any point in time. The valid actions depend on the current mode of the system. The user must always be made aware of what mode she is in, to know what constrained set of actions she can use. This is a much more challenging task for a non-visual user interface. In *Nomadic Radio* a simple vocabulary of commands is applied that to all messages, rather than unique commands for each view (see figure 4.2). These commands are processed to act appropriately on the media type of the message. For example parameters for spatial audio playback or synthetic speech feedback are changed in a similar manner.

Nomadic Radio adopted modeless interaction from its inception. This is based on related work by Lisa Stifelman on the moded vs. modeless interface in *VoiceNotes* [Stifelman92]. In *Nomadic* Radio, this ensures a scaleable approach as new information services or categories are added by developers or subscribed to by users in the future. Interaction with future services such as periodic weather forecasts, traffic reports, stock reports or recorded personal notes would be identical for the user.

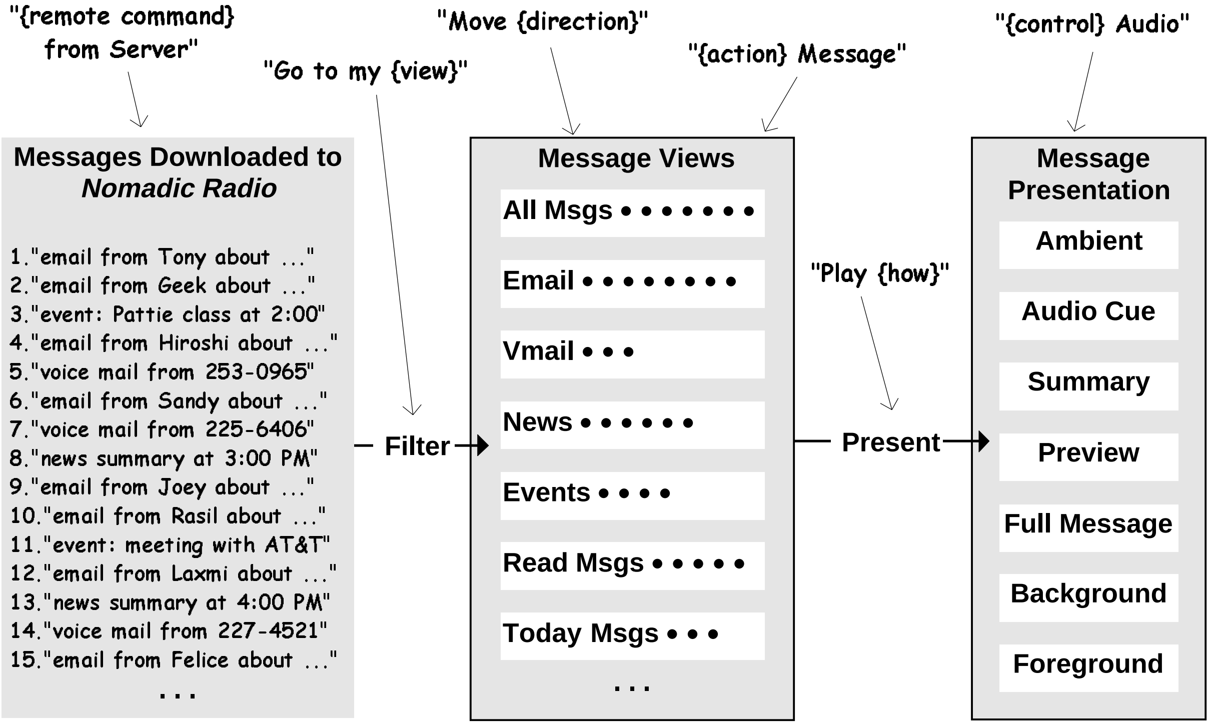

"{remote command} from Server"
"Go to my {view}"
"Move {direction}"
"{action} Message"
"{control} Audio"
Messages Downloaded to *Nomadic Radio*
1."email from Tony about ..."
2."email from Geek about ..."
3."event: Pattie class at 2:00"
4."email from Hiroshi about ..."
5."voice mail from 253-0965"
6."email from Sandy about ..."
7."voice mail from 225-6406"
8."news summary at 3:00 PM"
9."email from Joey about ..."
10."email from Rasil about ..."
11."event: meeting with AT&T"
12."email from Laxmi about ..."
13."news summary at 4:00 PM"
14."voice mail from 227-4521"
15."email from Felice about ..."
. . .
Filter
Message Views
All Msgs • • • • • • •
Email • • • • • • • •
Vmail • • •
News • • • • • •
Events • • • •
Read Msgs • • • • •
Today Msgs • • •
. . .
"Play {how}"
Present
Message Presentation
Ambient
Audio Cue
Summary
Preview
Full Message
Background
Foreground

Figure 4.2: Modeless design of interaction with messages in *Nomadic Radio*. All messages are filtered, browsed and presented in a consistent manner. General interface commands are valid in any message view.

4.3 Voice Navigation and Control

Speech provides a rich means for human communication and it can be effectively leveraged for interaction with devices in our environment (or wearables on our body). The special characteristics and affordances of speech must be carefully considered in designing voice-enabled applications for wearables.

4.3.1 Why use Voice Input for a Wearable Interface?

Wearable applications must be designed to be unobtrusive and responsive, as the user expects to use them casually and instantaneously in a variety of nomadic environments. Voice input provides a unique advantage in situations where the user needs hands-free interaction, ubiquitous access and direct control especially for small and infrequent transactions, as discussed below.

Convenient and Hands-free Access

On a desktop environment, speech cannot effectively replace input devices such as the keyboard or mouse (except for people with RSI-related disorders that inhibits their ability to type or use the mouse). Yet, it can provide an alternative channel of interaction when the user’s hands and eyes are busy elsewhere, without having to shift their gaze to the screen [Schmandt94b]. For nomadic access to information and interface control, speech provides a natural and convenient mechanism. Nomadic access includes use of telephony, access while driving, and on portable or wearable devices such as *Nomadic Radio*. In these situations, a hands-free approach may be essential, where user do not need additional input devices such as keyboards and has a direct interface to her wearable. Simple and reliable voice instructions can replace an awkward series of keyboard inputs or point and click actions using an impractical set of visual prompts.

Extent of Transactions on Wearables

Another key issue is the kind of transaction that the user is engaged in on her desktop vs. the wearable. The desktop is a rich environment where users spend a lot of time on longer transactions such as composing and editing documents/email. On a wearable, interactions can be structured as smaller transactions such as receiving notifications, listening and browsing messages, or communicating with people. Such transactions allow the interface to utilize speech input more effectively where a few phrases allow sufficient control to complete the transaction, and allow the user to focus on the task at hand.

Ubiquitous and Light-weight Access

Finally, speech input provides a ubiquitous means for accessing information and controlling processes on remote machines, simply by using a light-weight platform such as wireless microphones. In *Nomadic Radio,* we have provided local and remote speech access to support a variety of modular architectures such as wired wearable vs. distributed wireless (see chapter 7 on the Software Architecture) and interaction techniques such as push-to-talk and continuous listening (discussed below).

We will now consider the issues related to speech recognition, vocabulary design and voice-based navigation in *Nomadic Radio*.

4.3.2 Speech Recognition Technology

There are several speech recognition technologies in use today. One has to consider aspects such as speaker independence, continuous recognition and vocabulary size to determine the appropriate solution for specific applications. Additional considerations in selecting a recognizer include the performance, memory, barge-in capability, OS compatibility and ease of application integration.

Template Matching vs. Sub-Word Analysis

Most systems today rely on *template matching*. Here, a set of template or models describing each word to be recognized, are created using a representation of speech used by the recognized [Schmandt94b]. These templates form the recognizer's vocabulary, acting as reference models with which to compare spoken input. Pre-processing on spoken words determines word boundaries, and the most similar template is selected if the difference is minor enough to accept the word. In contrast, *sub-word analysis* represents the utterance as a series of discrete components called phonemes. The phoneme is the atomic particle of speech, the smallest element of sound that can maintain a meaningful distinction in human language. For recognition, a sub-word analysis reduces the auditory pattern (a combination of noise and human speech) to a string formed from the letters in a phonetic alphabet. The system then decodes the message by matching this phonetic transcription with that of the phrases stored in its vocabulary.

Speaker Dependence vs. Speaker Independence

In general, template-matching is speaker dependent. Therefore, each individual user is required to train the system on her speech for all the words in the vocabulary. This can be a lengthy process that may be undesirable for many casual users of an application. Although users of a wearable audio application may be more willing to train such a recognizer, as they are the primary users and have a vested interest in using the application on a daily basis. A sub-word analysis is entirely speaker-independent since it relies on phoneme based representations of the speech. Hence it requires no explicit training by a user. This is especially useful in allowing iterative vocabulary design for a speech-centric application.

Discrete Word Recognition vs. Continuous Speech

Template matching systems generally recognize discrete words only. Phrases and sentences are handled by treating each utterance as a single word. Here, each part of the phrase must match the template exactly or recognition suffers. Such a discrete word recognition approach works well on dictation applications where users are forced to provide unnatural input consisting of short utterances separated by distinct silences. In continuous speech recognition systems, phonetic sub-units within words are identified, allowing users to speak in a natural and conversational style. This approach analyzes phrases and identifies significant content, so the system tolerates utterances that are not an exact match to the expected phrase. Yet, the users must still speak within a restricted grammar defined by the application.

4.3.3 Integrating Speech Recognition in Nomadic Radio

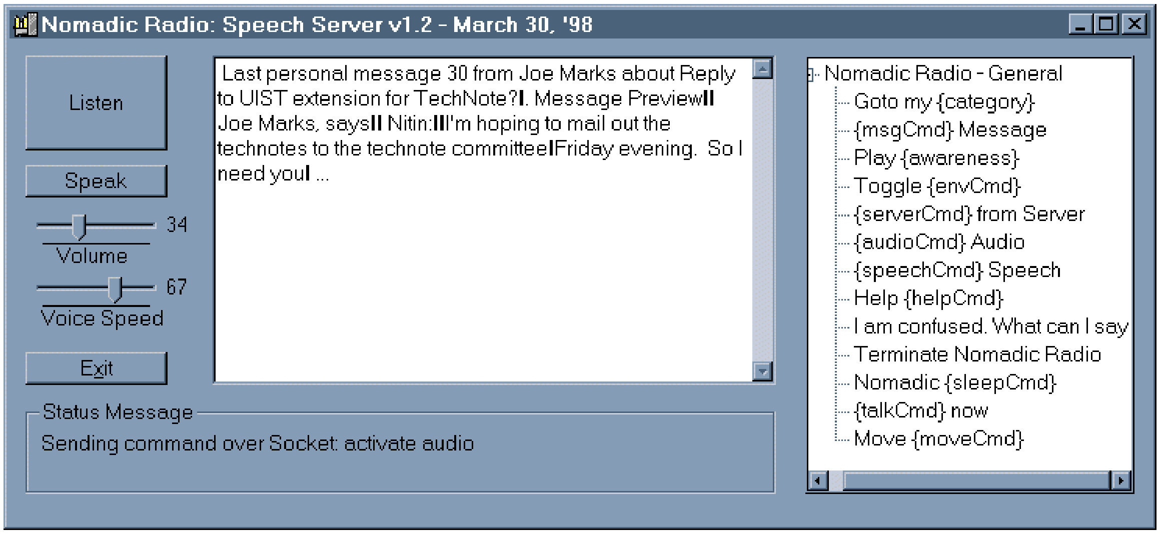
Voice input in *Nomadic Radio* is primarily used for browsing messages and controlling high-level interface actions. In this application, there is no need for dictation or to let the user add new words to customize the vocabulary. A continuous speech speaker independent recognizer is preferable since we wish to provide natural spoken interaction to the user with no need for extensive training. Independent-speaker recognizer allows the wearable application to be used easily by others as needed.

Selecting a Speech Engine for Nomadic Radio

We chose to use the *Watson* SDK (software development kit) from AT&T corporation [AT&T97]. At the time, the SDK was in Beta (version 2.0) whereas recently it has been released as a full product (*Watson* version 2.1). The *Watson* product is an integrated, automatic speech recognition (ASR) and text-to-speech (TTS) synthesis system that complies with the Microsoft Speech API specification (SAPI). The *Watson* ASR Engine utilizes phoneme-based sub-word analysis and hence supports speaker independence and continuous speech recognition. *Watson* runs on 32-bit PC platforms using Windows 95 or Windows NT. It requires a Pentium processor (75 MHz or faster), 16 MB of installed memory and 10 MB disk space. With a grammar of less than 100 active phrases, 2 Mb of RAM, and 50% of the CPU, *Watson* responds in real time. In theory, these requirements should not pose much load on current wearable PCs (whereas other recognizers have much higher requirements). Yet, with several applications running on the wearable, additional resources will be required. One advantage of using *Watson* is that the SDK was compatible with existing 32-bit development tools on PCs. This provided us an open speech API (application programming interface) for developing our own networked speech module operating on a PC-based wearable platform (see figure 4.3).

Recognizer-Independent Network Implementation

The networked speech module was interfaced with *Nomadic Radio* using a sockets protocol, such that it could be run either on the local wearable PC or operated on a remote networked PC via a wireless microphone (this architecture will be discussed in greater length in chapter 7). A sockets protocol permits the application to be independent from the recognizer shielding it from new Speech Engine releases or software updates to the module. This approach allowed us to test the Speech Recognizer and the *Nomadic Client* Java application independent of each other by sending/receiving messages to them via simulated sockets clients. As an added benefit, a sockets implementation inherently permits easy integration of a different Speech Recognition Engine (for use on an alternative OS platform), without any change to the application code itself.



Last personal message 30 from Joe Marks about Reply to UIST extension for TechNote?I. Message PreviewII Joe Marks, saysII Nitin:III'm hoping to mail out the technotes to the technote committeeIFriday evening. So I need youI ...

Sending command over Socket: activate audio

Figure 4.3: The Speech Recognition and Synthesis module running as a server on the local Wearable PC in a push-to-talk mode or on a remote networked machine in continuous speech mode.

Push-to-Talk vs. Continuous Monitoring

Voice recognition is provided via two different operational modes: *push-to-talk* and *continuous monitoring*. In noisy environments, a push-to-talk strategy allows users to explicitly direct commands to the system or deactivate recognition completely. A time-out setting allows the user to press the listen button and start speaking. The system automatically detects end of the utterance and stops listening. Here, the system prompts the user with spoken feedback (*"Say that again?"*) or if it does not recognize a phrase when the user presses the listen button. A *push-to-talk* mode is also necessary for Wearable PCs without full-duplex audio support. Here the speech module notifies the *Nomadic Client* to silence all audio playback when the system is listening, and reactivate audio once it has heard an utterance.

During *continuous monitoring*, the system always listens to the user’s speech and only sends the command to *Nomadic Radio* if it has confidence in the recognized phrase. The user will only be notified when a command is recognized via spoken feedback (in high interruption mode) or audio cues (in low interruption mode), rather than annoying prompts triggered by noise or unintentional speech in the environment. The user can explicitly place the system in *listen* or *sleep* mode using the *trigger phrases "Nomadic Wake-up"* and *"Nomadic Sleep"*. While listening, the system tries to recognize any commands heard and will notify the user via spoken feedback (in high interruption mode) or audio cues (in low interruption mode) *only* when it has confidence in a recognized phrase. *Continuous monitoring* supports the ability for users to *barge-in* with spoken commands while the system is speaking or playing an audio stream. *Barge-in* makes it possible for the wearable application to be responsive to users in a natural way, without much effort on their part. Generally *barge-in* capability requires a specialized hardware solution (even in the current *Watson* 2.1 release). Our networked implementation of the speech module implicitly permits *barge-in* since the speech recognizer runs on a remote machine with a dedicated audio card.

4.3.4 Nomadic Radio Vocabulary Design

In *Nomadic Radio,* the vocabulary is structured into 12 meta-commands each of which support a unique set of modifiers, such as *"Go to my {email | news | calendar | voice-mail}," "Move {forward | back}"* or *"{play | stop | pause | slow-down | speed-up} Audio".* These command definitions are shown in figure 4.1. The user can say *"Help {command}"* for specific spoken instructions or ask *"I am confused. What can I say?"* to hear a overview of all help commands. As mentioned earlier, *Nomadic Radio* utilizes a *modeless* interface for unified messaging such that all voice commands are always valid within each category. Hence, commands like *"move back"*, *"play message preview"* or *"remove this message"* can apply to email, voice mail, news or calendar events. The following are some of the considerations taken for designing and selecting a vocabulary for *Nomadic Radio*:

Selecting Single Word vs. Phrase Commands

In *VoiceNotes* [Stifelman92], single word commands are used due to the nature of the speech recognizer. The discrete word recognizer requires users to pause between words. Commands such as *"Play {listname}"* are eliminated in preference to simply *"{listname}".* In *Nomadic Radio,* a continuous speech recognizer permits users to speak longer phrase commands more naturally. Commands such as *"Go to my {view}",* provide a more intuitive conversational approach to voice input. In addition, such commands reduce the recognition errors since shorter words are more difficult to distinguish from one another than longer phrases.

Avoiding Acoustic Similarity

Words that sound similar are likely to be more difficult for the recognizer to distinguish. Commands like *"read message"* and *"delete message"* used originally, were easily confused by the recognizer. In this case, the command *"delete message"* was subsequently replaced by its semantic equivalent *"remove message".* The vocabulary was redesigned after several usage iterations to select commands with minimal acoustic similarity. This approach reduces errors, yet overall accuracy is also influenced by noise in the environment and stress in the speaker’s voice.

Avoiding Coarticulation

Short phrases require users to speak quickly without pausing and this can cause coarticulation. Coarticulation is the process whereby the pronunciation of a phoneme changes as a function of its surrounding phonemes [Schmandt94b]. Interaction between coarticulation and lexical stress in a speaker’s voice causes unstressed syllables to change more easily. This leads to *vowel reduction* where the vowel of the unstressed syllable is shortened and turned into a schwa. For example, in commands *like "go to email"* or *"remove message"*, the *"e"* in *"email"* and *"remove"* becomes unstressed reducing the command heard by the recognizer as *"go to-mail"* and *"remoo-message"*. Such commands were modified to longer phrases that could be spoken naturally without the coarticulation effects, i.e. *"go to my email"* and *"remove this message"*. Note that the words *"my"* and *"this"* allow the speaker to slow down and pause naturally while speaking, making it easier to articulate the phrases without stress in the voice.

Reducing the Vocabulary Size

A large vocabulary size can increase acoustic similarity and reduce the user's ability to recall correct commands for a specific task. In *Nomadic Radio*, the vocabulary consists of *Message Actions* which apply to individual messages and *Operational Commands* used to control general interface functions, independent of specific messages. *Message Actions* include only five general commands, i.e. *"Go to my {view}"*, *"Move {direction}"*, *"{action} message"*, *"Play {how}"*, and *"{control} audio"*. The remaining nine commands are *Operational* and generally used less frequently, such as *"Nomadic Wake-up"* or *"Toggle Sleep"*. By creating meta-commands with modifiers, the vocabulary size is constrained, making the vocabulary easier for users to learn and recall. New actions can also usually be added as modifiers to the meta-commands in the existing vocabulary.

Command Synonyms

In many cases, more than one command phrase can be used to perform the same action. For example, *"Read this message"* and *"Play audio"* can read/play a text or audio message. In a mixed-media messaging system, actions on either media type should apply to the other (text or audio) to avoid confusion and simplify the user's interaction. Similarly, *"Stop talking now"* and *"Toggle speech feedback"* both have an identical function, but are supported since the user may expect to speak such commands in either context (toggle controls a variety of environmental settings). The user's overall goals must be supported by allowing predictable actions on contextually or semantically similar commands.

4.3.5 Voice Navigation Techniques

The following techniques allow user to browse text and audio-based messages using voice commands. A key issue is playing the target message selected by the user while messages are temporally scanned.

Browsing

Users can navigate among different views of the messages by saying "Go to my {view}". This provides a fast and intuitive mechanism for filtering messages. When a new view is generated, the system lets the user know the number of messages in the current view and a summary of the last message in the view is played. The user can move sequentially within the view by saying *"move forward"* or *"move back"*. Yet, such fine movements can get tedious if done frequently [Stifleman93] (these functions should be handled via tactile input). The user can also jump to messages at the beginning or end of the view via the commands *"first message"* or *"last message"*. If the user is on the last message, moving forward retains the current position and the user is told *"Already on last message"*. Like *VoiceNotes*, the beginning and end of views in *Nomadic Radio* act as anchors rather than drop-off points. If the user selects a view with no messages, she is returned to the overall view containing all messages. When a new message is received, it is inserted to the end of the current view and the user position is shifted to that message.

Scanning

The user can automatically browse through all messages in a view by *scanning* i.e. saying *"scan messages"*. This presents each message for a short duration, beginning with the last message in the view, moving backwards until all messages are played. The duration is selected based on the user's selected presentation level, i.e. audio cue, summary, preview or full message (discussed in section 5.5). The scan function works identically on both text and audio messages in any view. For audio messages, spatial foregrounding techniques are used (see section 4.7.6). For text messages, spoken via synthetic speech, there is a 1 second delay (period of silence) between messages during scanning. For audio messages, spatial audio allows a slight simultaneous overlap between messages. Once the user hears a message of interest or wants to stop scanning, issuing any command will deactivate scanning.

Temporal Target Windows

During scanning, when a user interrupts the playback to hear the current message, there is a problem of selecting the exact "target message". Depending on when the system hears the interruption command, one of three possible messages will be selected - the desired target message, the previous one or one following the target [Stifleman92]. This is caused by two factors: (1) delay in the user, who interrupts a few seconds after the desired message is first heard and (2) latency in the system's response time, while receiving the spoken command from the remote speech server. Generally, the user's command follows (lags behind) the playback of the target message. This can cause the wrong message to be selected for playback. Muller and Daniel [Muller90] suggest a "partially overlapping temporal window" to select the correct target. In *Nomadic Radio*, the temporal target window for a message being scanned, extends 2 seconds after it has finished playing (see figure 4.4).

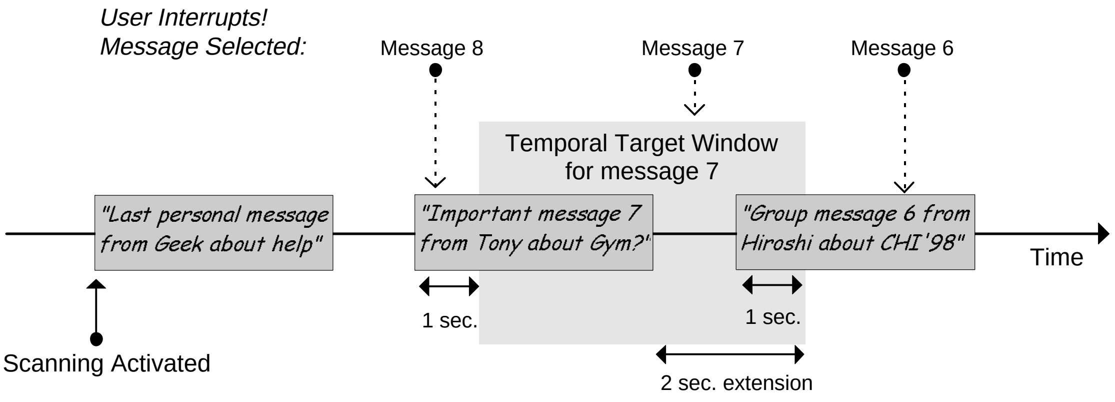


Figure 4.4: Scanning email messages and selecting the current message within the temporal target window.

The 2 second extension in the window takes into account a 1 second silence and playback of an audio cue, before the next message is presented. This approach works well and seems to provide the appropriate intuitive target selection based on iterative testing by the author.

4.3.6 Problems with Voice Input on Wearables

Several characteristics of speech recognition and spoken commands in noisy and social environments make the use of voice input on wearables ineffective or simply awkward in those situations. In this section, we discuss these problems and consider potential solutions.

Unreliable in Noisy Environments

In noisy environments, the accuracy of recognition is seriously degraded and the interface can be less responsive. Directional microphones and noise cancellation[14] techniques ease the problem to some extent. In addition, if users are stressed or frustrated they will not articulate spoken commands clearly and this too will affect the recognition accuracy.

Adopting New Social and Cultural Conventions

Using speech recognition on a wearable device typically requires use of body-worn microphones. Most social and cultural conventions assume that it is awkward for people to be speaking to themselves, especially with no prior warning (unlike taking calls on a cell-phone). It can be confusing whether the user is addressing her wearable or the person next to her in an elevator or meeting room. An explicit push-to-talk button that produces an audio cue heard by others and a continuous visual indicator (flashing light on the wearable) can reduce some of the confusion. Users speaking during a meeting or lecture can be distracting to others in the room (unless they whisper). Over time, people adopt new social conventions or simply get accustomed to such technologies, as they have with people using cellular phones in social environments. However, during the early phase of adoption, many users will be less inclined to use speech on wearables in public places.

Lack of Privacy, Security or Confidentiality

In a social environment, speech input poses a number of additional problems. Users will not feel comfortable speaking the names of people they received messages from to retrieve those messages. Speaking passwords or confidential information (financial or medical transactions) would be undesirable near coworkers. Therefore the application must be designed to minimize such transactions via speech or simply provide alternative means of input.

[14] *Good directional microphone design provides some form of noise cancellation. Yet active noise cancellation requires pre-sampling the level of noise in the background and subtracting it from the user's speech (a much more difficult problem).*

Must Learn or Easily Acquire Vocabulary

Unlike on-screen buttons, speech commands on a non-visual application must be recalled by the user. The vocabulary can be designed to provide intuitive commands, yet the user must be made familiar with their syntax and extended functionality over time. In many cases, the user may not recall the right command to speak, and either needs to inquire the application (use of help commands in *Nomadic Radio*) or needs an alternative means for accomplishing the task easily.

Speech is Slow and Tedious for Repetitive Commands

Excessive speech interaction can be tedious especially if the user must repeat the same command in a noisy environment. Tasks requiring fine control such as "faster audio" or repetitive input such as "move forward", make speech input awkward "faster, faster ...." [Stifleman93]. Such tasks are better accomplished using tactile input.

We will now consider an alternative to speech in situations where speaking is less desirable or recognition is simply not feasible. Tactile input using a wireless keypad provides a potential solution that can be coupled with speech to provide a responsive and unobtrusive wearable interface.

4.4 Augmenting Voice Input with a Tactile Interface

As spoken commands on a wearable are not feasible in many situations, tactile input provides a means for users to control *Nomadic Radio* in a discreet and unobtrusive manner. Speech input is more suitable for "coarse" navigation commands [Stifelman92] such as *"Go to my voice mail,"* whereas tactile input is better suited for "fine" grained commands such as *"move back"* or *"speed-up audio"*. Tactile input for *Nomadic Radio* is implemented via a numeric keypad connected to a networked Toshiba *Libretto* 50CT mini-notebook PC (see in figure 4.5). This allowed a rapid means for prototyping tactile input using existing hardware infrastructure. The numeric keypad provides a simple one-handed interface onto which a number of *Nomadic Radio* functions can be mapped. One benefit of this approach is that it permits light-weight implementation of the system where the user only wears a wireless microphone and speakers while carrying the wireless keypad for additional input. This works well in an inter-office environment with a Wave-LAN infrastructure. On a wearable configuration, this functionality should be mapped to a special purpose wired tactile input device customized for *Nomadic Radio,* in the future.

Figure 4.5: A numeric keypad connected to a networked Toshiba *Libretto* 50CT mini-notebook PC. This is useful for providing tactile feedback when speech is not feasible.

4.4.1 Mapping Voice Commands to a Tactile Interface

The goal of providing tactile input was to allow a sufficient subset of the frequently spoken commands to be substituted by the keypad. Most of the *Message Actions* and a few *Operational commands* are mapped to the numeric keypad. The mapping scheme (shown in figure 4.6) is designed and iterated several times to find a seemingly intuitive mapping between the existing keys on the keypad and the commands. Critical functions, i.e. *"Remove Message"*, *"Toggle Sleep"* and *"Scan Messages"* are only accessible when the *num-lock* on the keypad is active. This prevents accidental activation by the user.

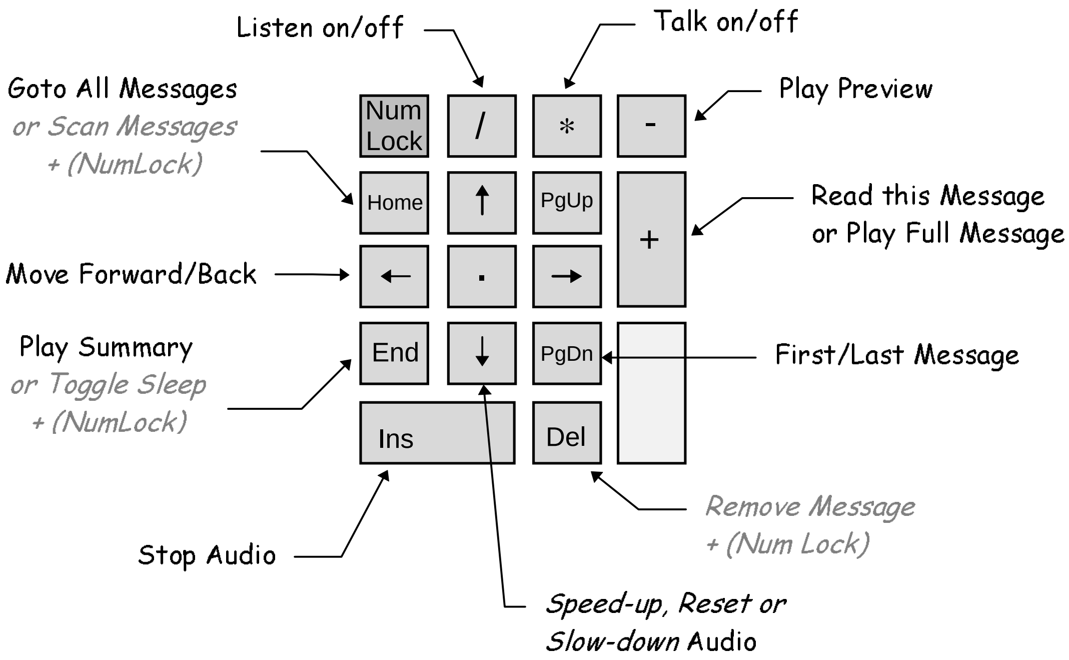
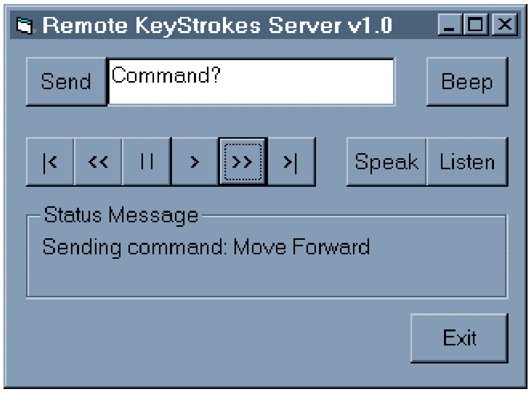


Figure 4.6: Mapping a subset of commands to the keypad. Note that the user must switch to num-lock mode to remove messages or activate sleep mode.

4.4.2 Networked Implementation

A networked application, the *KeyStrokes Server*, running on the *Libretto* (see figure 4.7) monitors the user's key strokes and sends the appropriate command to the Nomadic Client (the audio application running on the wearable) via a sockets protocol. Buttons on the application are used for testing the high-level functions if the numeric keypad is disconnected. Arbitrary commands can also be sent to the Nomadic Client (for testing) by typing them into its command window.



Command?

Status Message

Sending command: Move Forward

Figure 4.7: The networked *KeyStrokes Server* application running on the *Libretto*, that sends the user's key-presses to the Wearable Nomadic Client.

4.4.3 Auditory Confirmations

The application plays an audio cue (a subtle beep sound) to assure the user that a command was actually sent over the network. Lack of audio cues remind the user when the *num-lock* key was not pressed for commands like *"Remove Messages"*. Finally, an audio cue is also heard if the application loses its connection with the *Nomadic Client*. Audio cues can be turned off by the user, if desired.

4.4.4 Hybrid Interaction

The addition of tactile input provides a hybrid speech-button interface [Stifelman92] that allows both modalities to work well in conjunction with one another. Similar commands issued by either speech or tactile input provides the same feedback to the user to maintain consistency. Yet, since button input is more reliable than speech, explicit confirmations are reduced from spoken speech feedback to audio cues. The user has the ability to use a combination of speech and buttons effectively for a sequence of actions. For example, the user can switch to a new view by saying *"Go to my Calendar"* and then use the forward/back buttons to quickly browse the events in her calendar. In addition, actions initiated in any one modality should be allowed to be terminated by another. For example, playing a message by saying *"Play Summary"* can be terminated by pressing the *"Stop Audio"* key. Such a hybrid approach, provides the most effective means for interaction with a temporal and non-visual interface. Over time, users will adopt preferred modalities for specific functions.

4.5 Spoken Feedback via Synthetic Speech

4.5.1 Recorded vs. Synthetic Speech

Pre-recorded voice prompts provide a natural means for conveying feedback, but this constrains the spoken vocabulary of the interface. Synthetic speech is necessary for applications where prompts must be dynamically generated and textual information such as email or calendar events is conveyed to the listener [Klatt87]. Synthetic speech also provides easier development and maintenance, when new prompts are added to the application without any need to explicitly pre-record prompts (and find the original speaker). The perception of synthetic speech does impose greater demand on short-term memory more than recorded speech [Luce83]. Difficulty in listening to synthetic speech may interfere with the listener’s tasks and even remembering the information being spoken [Schmandt94b]. Such prompts must be designed carefully to convey important information using both concise and comprehensible utterances.

4.5.2 Design of Speech Prompts in Nomadic Radio

Synthetic speech is most appropriate if relatively short prompts are used. This permits faster interaction and requires the listener to retain less information in working memory. In *Nomadic Radio*, speech prompts are designed to be brief, yet need to convey sufficient information to be useful. Many of the prompts use a conversational style rather than being too abrupt. This provides natural sounding feedback that is easier for users to listen to. There is a clear trade-off between conversational vs. short prompts [Hanson96]. In general, the user should be allowed to set the level of preferred feedback (discussed in the next section). In addition, the *barge-in* capability provided in *Nomadic Radio* allows the user to interrupt spoken feedback with a voice command or simply terminate it (using a button) if she has understood the general context. Figure 4.8 shows examples of the different speech prompts used in *Nomadic Radio*.

Informative Feedback and Notification

NR: *"Nomadic Radio launched. Loading your messages, Nitin. Please wait ..."*

NR: *"New voice message from 253-0352."*

Explicit Confirmation

Nitin: *"Nomadic Wake Up!"*
NR: *"Ok, I'm Listening."*

Implicit Confirmation

Nitin: *"Go to my calendar."*

NR: *"Nitin, you have 3 scheduled events today."*

Error Notification

NR: *"Already on last message!"*

NR: *"Disconnected from remote speech server"*

Figure 4.8: Design of speech prompts for feedback, confirmations and notifications.

The system provides the user feedback when it needs to load all messages during startup and lets the user know when the messages are loaded. This provides assurances to remain patient while the process is running. When the system recognizes a speech command, several forms of feedback are provided, ranging from audio cues to implicit and explicit confirmations. Echoing the spoken command to novice users is the most explicit form of confirmation. Such feedback gets tedious with continued usage, so a reduced form of confirmation is provided via audio cues (discussed in the next section) or by indirectly implying the correct action taken in the information spoken. Figure 4.8 shows an example of an implicit confirmation when the user requests her calendar events. Instead of echoing *"Going to your calendar"* the system simply says *"you have 3 scheduled events today"* indicating it has switched to the calendar view.

4.5.3 Scaleable Feedback and Message Notification

Several levels of feedback allow the system to inform the user of notifications or confirmations of her actions. Feedback could be scaled back based on experience with the system, confidence in the speech recognition results, as well as the level of interruptability desired. In *Nomadic Radio*, four main levels of feedback can be selected by the user (via spoken commands) or automatically by the system via dynamic operational modes: (1) explicitly set the system to provide maximum feedback and echo all spoken commands by setting "High Interruptability" (2) allow implicit confirmations by letting the system stay in talk mode, (3) hear only audio cues for notification, and (4) hear only ambient sounds for awareness (discussed in the next section). The system also dynamically scales the feedback based on the usage level and level of conversation in the environment (discussed in section 5.7 - *dynamic operational modes*). Notifications for messages are scaled from audio cues to summary and preview modes (discussed in section 5.5). See figure 4.9. for examples of how attributes are spoken when messages are summarized. It is interesting to consider the effect of using synthetic speech with increased pitch or volume to convey urgency or decreased pitch to aid comprehension and emphasize the subject [Marx95].

"Event 1. Garden Lunch at 12 P.M. for 60 minutes."

"Most important message 2 from Geek about Reply to thesis draft."

"Short Message 3. Voice message from 253-4086"

"Long Message 4. News Summary at 2 P.M."

Figure 4.9: Message summaries generated for calendar, email, voice mail, and news.

4.5.4 Synchronizing Synthetic Speech with Audio Cues and Spatial Audio

Finally, to provide effective feedback to the listener, the timing of the spoken speech feedback is paced with adequate periods of silence. It is also synchronized with audio cues and the audio streams to be played. The user can hear the audio cues, voice messages and spoken notifications in the correct order and comprehend each one before it has finished playing. When the system needs to convey a timely notification via synthetic speech, it will temporarily pause all currently playing audio sources and resume them after it is done speaking. An alternative approach would render the synthetic speech in the foreground or background while allowing the current audio sources to continue playing. This approach was not utilized since simultaneously playing synthetic speech requires saving it to a file and rendering it as a localized sound source within the spatial audio environment. This produces an added delay in hearing the spoken speech, which was found to be undesirable for most feedback and notifications.

4.6 Awareness and Notification via Auditory Cues

Excessive speech feedback is tedious and can slow down interactions on a wearable. Synthetic speech can be distracting to listeners while they are performing other tasks or having conversation. On the other hand, non-speech audio in the form of *auditory icons* [Gaver89] or cues provide such feedback via short everyday sounds. Auditory cues are a crucial means for conveying awareness, notification and providing necessary assurances in a non-visual interface. Four different sets of auditory cues are designed for distinct aspects of feedback and information conveyed in the *Nomadic Radio* interface.

4.6.1 Feedback Cues

Several types of audio cues indicate feedback for a number of operational events in *Nomadic Radio*: *(1) Task completion and confirmations* - button pressed, speech understood, connected to servers, finished playing or loaded/deleted messages. *(2) Mode transitions* - switching categories, going to non-speech or ambient mode. *(3) Exceptional conditions* - message not found, lost connection with servers, and errors.

4.6.2 Priority Cues for Notification

In a related project, email glances [Hudson96] were formulated as a stream of short sounds indicating category, sender and content flags (from keywords in the message). In *Nomadic Radio,* message priority inferred from email content filtering (see section 5.6 on contextual cues) provides distinct auditory cues (assigned by the user) for group, personal, timely, and important messages. In addition, audio cues such as telephone ringing indicate voice mail, and an extracted sound of a station identifier indicates a news summary. The size of the message is depicted by changes in the background ambient sound, which also acts as an implicit notification that the message is being loaded.

4.6.3 VoiceCues for Identification

VoiceCues[15] represents a novel approach for easy identification of the sender of an email, based on a unique auditory signature of the person. *VoiceCues* are created by extracting a 1-2 second audio sample from the voice messages of callers and associating them with their respective email login. When a new email message arrives, the system queries its database for a related *VoiceCue* for that person before playing it to the user as a notification, along with the priority cues. *VoiceCue*s have been found to be a remarkably effective method for quickly conveying the sender of the message in a very short duration. This reduces the need for synthetic speech feedback which can be distracting. In the future, the inferred priority or urgency of the message could be depicted by increasing the pitch of the *VoiceCue*.

4.6.4 Ambient Cues for Background Awareness

In *ARKola* [Gaver91], a audio/visual simulation of a bottling factory, repetitive streams of sounds allowed people to keep track of activity, rate, and functioning of running machines. Without sounds people often overlooked problems; with sounds these were indicated by the machine's sound ceasing (often ineffective) or various alert sounds. The various auditory cues (as many as 12 sounds play simultaneously) merged as an auditory texture allowed people to hear the plant as a complex integrated process. Background sounds were also explored in *ShareMon* [Cohen94], a prototype application that notified users of file sharing activity. Cohen found that pink noise used to indicate %CPU time was considered obnoxious, even though users understood the pitch correlation. However, preliminary reactions to wave sounds were considered positive and even soothing.

In *Nomadic Radio*, ambient auditory cues are continuously played in the background to provide an awareness of the operational state of the system and ongoing status of messages being downloaded. The sound of flowing water provides an unobtrusive form of ambient awareness that indicates the system is active (silence indicates sleep mode). This sort of sound tends to fade into the perceptual background after a short time, so it does not distract the listener. The pitch is increased during file downloads, foregrounding the ambient sound momentarily. A short e-mail message sounds like a splash while a two minute audio news summary is heard as faster flowing water while being downloaded. This implicitly indicates message size without the need for additional audio cues and prepares the listener to hear (or deactivate) the message before it becomes available.

[15] *The idea for using VoiceCues came about during a spontaneous discussion about email notifications with Chris Schmandt. I must admit I was much more skeptical about their effectiveness until they were integrated in Nomadic Radio, and I began using them for notifications on a daily basis. In one instance, I heard the voice of an old friend in the Netherlands (a VoiceCue from her email), while we were talking about her in my office; an unusual and serendipitous experience indeed.*

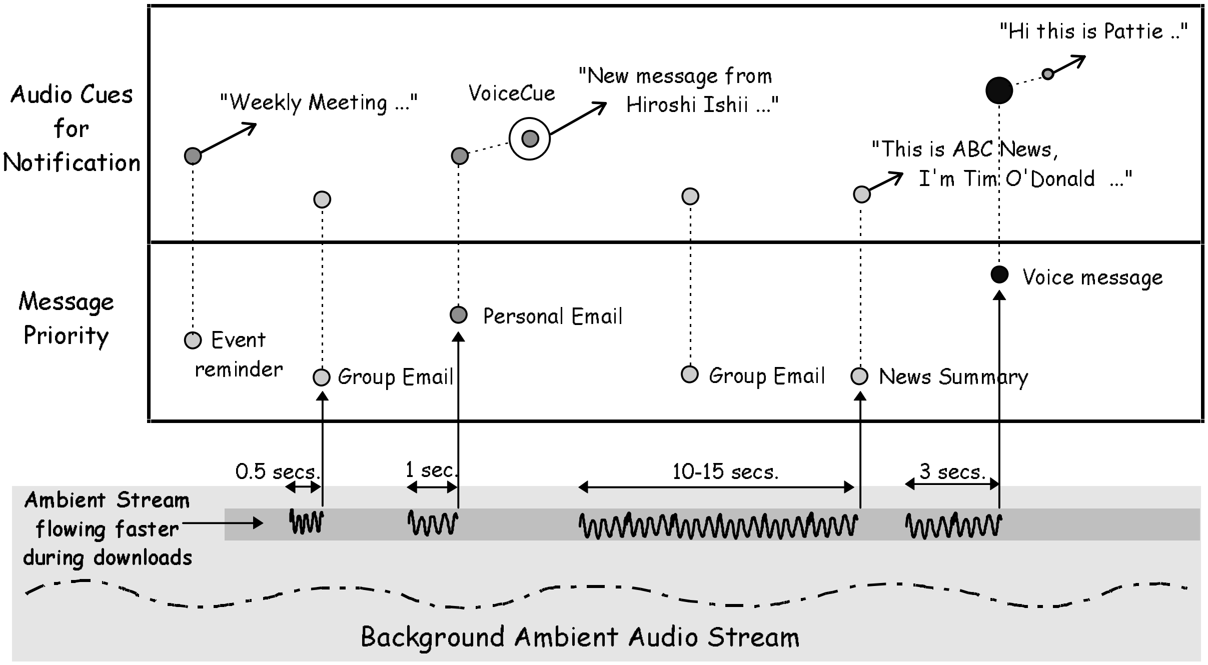

Audio Cues for Notification
"Weekly Meeting ..."
VoiceCue
"New message from Hiroshi Ishii ..."
"Hi this is Pattie .."
"This is ABC News, I'm Tim O'Donald ..."
Message Priority
Event reminder
Group Email
Personal Email
Group Email
News Summary
Voice message
0.5 secs.
1 sec.
10-15 secs.
3 secs.
Ambient Stream flowing faster during downloads
Background Ambient Audio Stream

Figure 4.10: Ambient auditory stream speeded-up while downloading incoming messages. Audio cues indicate priority and *VoiceCues* identify the sender. A few seconds after the audio cues, the message content is played as synthetic speech or spatial audio.

All priority cues and *VoiceCues* are played in specific locations around the listener based on the time of arrival of the associated message. This provides a strong correlation with the audio stream of the message during notification and browsing. The ambient sound is positioned at the center of the listening space and played in the background relative to all other sounds. Techniques for localization of sound and design of the spatial audio display are discussed in the next section.

4.7 Simultaneous and Spatial Listening

A spatial sound system can provide a strong metaphor for listening to audio by placing individual voices in particular spatial locations (as discussed in section 1.3.5 on the benefits of simultaneous listening). The effective use of spatial layout can be used to aid auditory memory. The *AudioStreamer* [Schmandt95] [Mullins96] detects the gesture of head movement towards spatialized audio-based news sources to increase the relative gain of the source allowing simultaneous browsing and listening of several news articles. Kobayashi introduced a technique for browsing audio by allowing listeners to switch their attention between moving sound sources that play multiple portions of a single audio recording [Kobayashi97]. An audio landscape with directional sound sources and overlapping auditory streams (*audio-braiding*) provides a listening environment for browsing multiple audio sources easily [Maher97].

4.7.1 Why use Spatial Audio in Nomadic Radio?

We will first consider some key reasons why spatial and simultaneous listening can be beneficial for notification and browsing of messages on an audio-only wearable system.

- On a hand-held or wearable audio system, lack of a visual display restricts the amount of information that can be easily conveyed at any time. "Spatial and perceptual streaming cues can help in presenting a high bandwidth information to the user by displaying multiple streams of information simultaneously." [Arons92]. *Nomadic Radio* allows simultaneous audio streams.
- While browsing audio messages, such as voice mail and news summaries, listeners need to find the messages (or segments of the audio) of interest in a time-efficient manner. Listening to simultaneous audio streams is not unlike visually scanning multiple objects in a graphical display.
- In a non-visual display, as messages are heard throughout the day, their spatial location can be a good indicator for allowing listeners to easily retrieve messages. Spatial memory can potentially also aid listeners in recalling the sender, urgency or context of the message when heard from a previously known location. We will consider how a *scaleable* spatial audio display can be designed, primarily based on the temporal attributes of incoming messages.
- In *Nomadic Radio*, several scaleable notifications for incoming messages (via audio cues and speech feedback) are provided to the listener. If the user is currently listening to an audio stream, new notifications are played in the foreground while the current audio stream is temporarily placed in the background. This allows a fluid listening experience without abrupt interruption in the audio streams from incoming messages.
- When multiple simultaneous audio streams are available, using spatial audio can allow listeners to focus on a primary stream, while listening to others in parallel in the background. "The goal is to keep the speech signals identifiable and differentiable, so that the user can shift attention between

the various sound streams." [Arons92]. The user can easily switch between "overheard" audio streams to focus on a relevant message.

- In a future extension of the system, audio-based communication with multiple participants will be provided (spatialized audio-conferencing). Here, a spatial separation of the speaker's voices enhances the listening experience due to better segregation and spatial memory.

4.7.2 Perception of Spatial Sound

We will now consider the perceptual phenomenon that allows us to hear auditory events in a spatial manner and describe techniques for simulating synthetic 3D reproduction. Sound waves that reach a listener's eardrums are affected by the interaction of the sound wave with the listener's head, *pinnae* (outer ears) and ear canals. The composite of these properties are measured and captured as a head-related transfer function (HRTF). When the sound source is not equidistant form the ears, the signal arrives at each ear from a different direction and the HRTFs at each ear differ [Kendall95]. The position of the sound relative to the center of the listener's head is conveniently captured as a vector expressed in terms of two angles, *azimuth* and *elevation*, and one scalar, *distance*. (see figure 4.11)

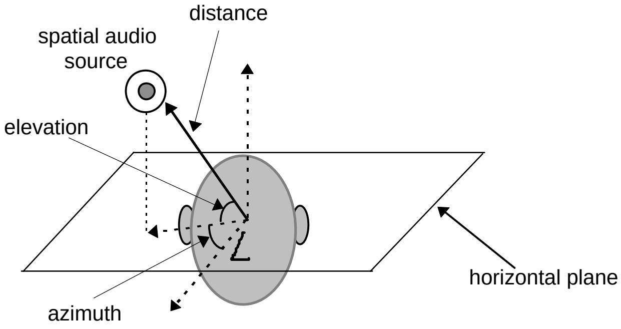


Figure 4.11: Position of a sound source relative to the head in terms of azimuth, elevation and distance

The sound arriving at the ear closer to the sound source will generally be more intense and arrive sooner than at the far ear. These differences between the two ears are called the interaural intensity difference (IID) and the interaural time difference (ITD). ITDs were thought to be responsible for localization at low frequencies due to phase ambiguities at higher frequencies. Spatial perception is strongly affected by these differences, however IID and ITD do not take into account the spectral shaping caused by the *pinnae*. The complex acoustic profiles of HRTFs provides a more accurate means for explaining localization and hence modeling synthetic spatial sound. Individual HRTFs will generally be similar for most people despite slight differences in their head size and *pinnae* (this difference will vary considerably for children) [Kendall95]. Elevation judgments and front-back differentiation are more likely to degrade with non-individualized HRTFs [Wenzel93], yet it is also clear that head movement plays a dominant role in resolving front/back confusions [Wallach40]. Overall, individuals generally localize better with their own HRTFs, however using a generalized set of superior HRTFs can sometimes improve their localization.

4.7.3 Integrating Spatial Audio in Nomadic Radio

In our implementation, the audio sources are rendered in the spatial environment of the listener using a Java interface to the *RSX 3D* audio API[16] [Intel97] developed by Intel. The perceptual audio models used in *RSX 3D* are based on a set of HRTF measurements of a KEMAR (electronic mannequin) by Bill Gardner at the MIT Media Lab [Gardner95]. The measurements consist of the left and right ear impulse responses from a loudspeaker mounted 1.4 meters from the KEMAR. The HRTF model allows real-time rendering of several monophonic sound sources positioned arbitrarily around the head and permits control of their elevation, azimuth, and distance cues.

In *Nomadic Radio*, we utilize the metaphor of *radio* to present information as *active broadcasts* delivered within the user’s listening environment. Several such broadcasts are presented simultaneously as spatialized audio streams to enable the listener to better segregate and browse multiple information sources. The key challenge is to design a scaleable spatial mapping for information. This should be augmented with effective auditory techniques that provide a unified listening experience.

On a wearable device, spatial audio requires the use of headphones or preferably shoulder mounted directional speakers using the *SoundBeam*. In noisy environments, there will be a greater cognitive load on the listener to effectively hear spatialized audio streams, yet a spatial display helps segregate simultaneous audio streams more easily. Here, the exact spatial location of the sound source is less important, but can provide cues about the message such as its category, urgency and time of arrival.

4.7.4 Techniques for Designing Spatial Audio Layout

Based on the issues discussed above, a number of techniques were considered for the design of spatialized audio streams. One goal was to maximize audible delivery and segregation of information as well as minimize the potential perceptual load on the listener. In addition, the display must be scaleable as many incoming messages must be positioned around the listener throughout the day. Spatial position must be assigned in an intuitive manner to allow the listener to easily browse the messages at a later time.

1. *Spatialized Audio Braiding:* Several audio streams can be spatialized around a listener's head by making one stream prominent and letting it decay as another stream begins to play. Multiple audio streams such as news articles or voice mail could be braided alternatively [Maher97] at pre-defined intervals based on the urgency of the information, content attributes or via user-controlled intervals. This technique was implemented in *Nomadic Radio* so listeners can *scan* all audio messages sequentially with braided playback (discussed in section 4.7.6).

[16] *http://developer.intel.com/ial/rsx/index.htm*

2. *Highlighting Audio-Streams:* When a new audio stream is introduced or an existing one must be "highlighted", it can be swiftly moved to the center of the user's listening space. Alternatively, the stream can remain in its existing location, and its gain and vertical position could be increased, while that of all other sounds are reduced for a short duration. This would allow listeners to bring the highlighted stream into perceptual focus and retain a memory of the audio stream, even as new streams are mixed in. *Foregrounding* in *Nomadic Radio*, is used for previewing voice messages and news summaries (discussed in section 4.7.6). For example, when a new voice message arrives, the audio fades in and gets prominent for a few seconds, while other sounds are reduced. If the user does not activate the message (press a button), the audio quickly fades away - allowing the user to at least recognize the voice of the caller and perhaps get a sense of the message content.

3. *Categorical Spatialization:* Listeners can define specific quadrants of their listening space as associated with specific categories of information. This is accomplished by allowing users to manually place sounds in spatial locations over an extended period of time. The system would then localize new sounds related to specific categories in similar quadrants of the listening space. Such an effect would allow users to utilize their spatial memory to quickly recognize the content type of the audio stream, without necessarily listening to the audio itself. This technique does not scale well as the number of categories increase, and any mechanism to position messages within each category would be arbitrary.

4. *Segregation via Pitch:* Bregman claims that stream segregation is better when frequency separation is greater between sound streams [Bregman90]. As an experiment, different sounds could be played at successively changing pitches (where a sound is highlighted briefly by normalizing the pitch), which could be used to permit rapid browsing of multiple audio streams simultaneously. Yet, compressed speech requires greater perceptual load, and as a result listening to more than one or two messages with varying pitches would not ease overall comprehension. This technique is explored for previewing messages as time-compressed spatial audio streams (see section 4.7.6).

5. *Segregation via Dynamic Movement:* As new audio streams are introduced to a listener's soundspace, they could be placed in "orbit" around the listener at varying or constant speeds [Kobayashi97]. The sense of motion could allow segregation of a new stream, especially if the user has a tactile handle on the stream, i.e. the user can grab a stream as it moves, and place it in a specific location, while the others remain fixed. This effect can be used to preview recorded news articles as passing by and fading away if not attended to, or have them placed into perceptual focus if the listener activates them. This technique may be useful to introduce new incoming messages, yet it has not been explored in *Nomadic Radio*. For incoming messages, retaining the message at a specific spatial position will allow listeners to recall the messages later. By moving the messages arbitrarily, the benefits of any such spatial recollection are reduced.

4.7.5 Spatial Messaging: Mapping Time to Space

Designing an effective spatial layout for a diverse set of messages requires consideration of their content and scalability issues. In *Nomadic Radio*, messages such as email, voicemail, news and traffic reports must be presented to the listener throughout the day. Each message category has a different level of urgency and messages within that category may themselves be considered timely or higher priority. One approach is to map messages in auditory space based on content category. Therefore, news is played in one direction and all voice messages in another. Yet, this does not scale well as many new messages arrive and the spatial layout only indicates categorical information.

Chronological Layout for Spatial Audio

Each message arrives at a different point in time, hence its date and time of arrival provide a unique parameter for spatial layout. A suitable approach is to utilize arrival time to position messages in chronological order around a listener's head (Figure 4.12). A spatial clock can permit messages arriving at noon to be positioned in the front and those at 3:00 PM on the right and so on. This approach provides a user-centered coordinate system for the spatial display, entirely based on the temporal attributes of the messages received.

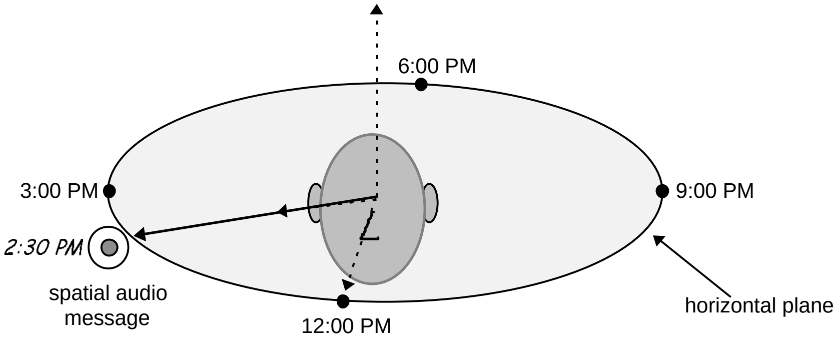


Figure 4.12: Localization of an incoming message around the listener. The message is localized based on its time of arrival, on a chronological (12-hour) spatial audio display.

A twelve hour clock does not scale well for messages arriving throughout the day. Messages arriving after 12:00 PM will overlap with existing ones from the AM. Auditory cues, played at the start of the message, can indicate AM or PM. Yet, an alternative is to use a twenty four hour clock which can represent messages for an entire day within the 360 ° audio space. Using such a metaphor, all messages arriving during the day occupy a unique position in the listening space. A 24-hour spatial mapping requires additional granularity by compressing the overall space. This reduces the spatial separation between messages arriving within a short time of each other. A minimum spatial separation is needed to discriminate between two sound sources. Early experiments found that localization blur ranged from approximately 1.5-5 degrees for various types of sounds presented sequentially [Blauert83]. For simultaneous presentation, the minimum recommended distance for accurate localization is 60 degrees

[Divenyi89]. In *Nomadic Radio*, spatial localization for individual messages primarily provides spatial memory and a general sense of message arrival time during notification and browsing. During simultaneous presentation, spatial position enables segregation and a means for focusing on the message of interest using techniques such as *foregrounding* and *scanning* (discussed in the next section).

Nomadic Radio permits both 12-hour and 24-hour spatial representations of the message space and the user can select one that provides a better spatial separation (12-hour mode) or temporal consistency (24-hour mode). See figure 4.13 for examples of 12 and 24-hour displays.

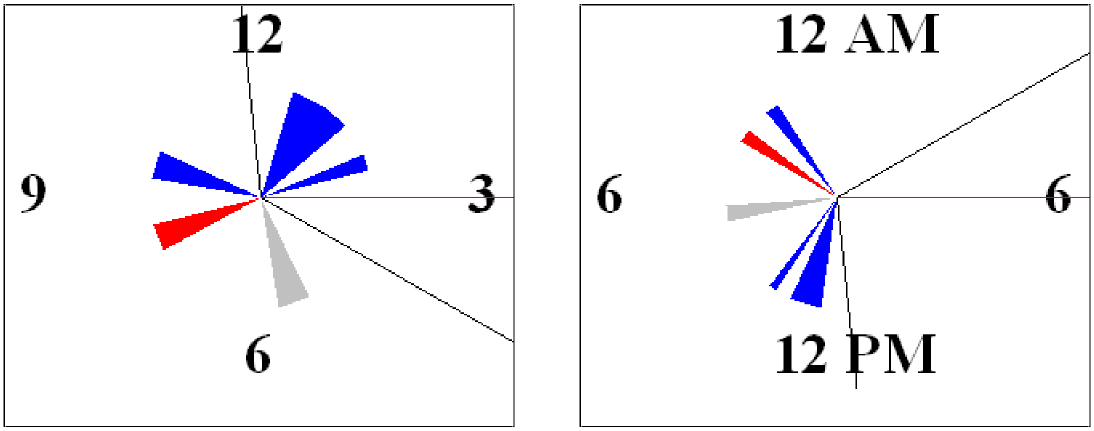


Figure 4.13: Spatial Layout of the voice messages using a 12 and 24-hour display mode

In either spatial layout, the listener can discern the approximate time of arrival based on the general direction that the message is heard. In addition, the message category determines the distance of the messages from the listener, indicating general importance of the category. For instance, all voice messages are positioned closer, whereas email messages and news summaries are placed further away (see figure 4.14) from the listener.

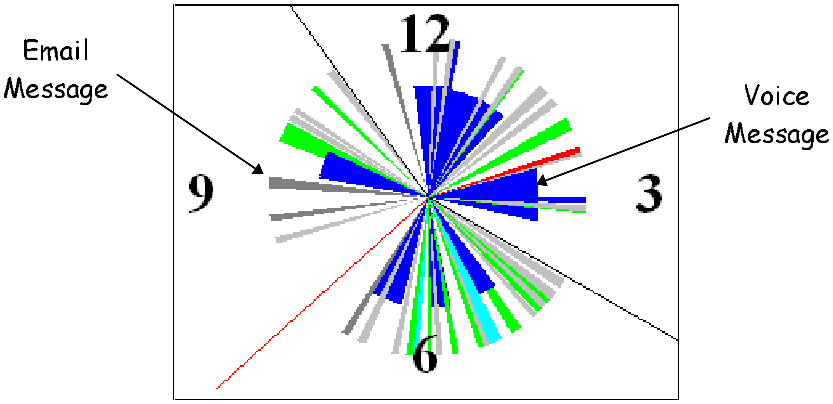


Figure 4.14: Spatial layout of voice messages and email (outer circle). Here the category of the message changes its distance from the listener (at the center).

Algorithm for Mapping Time to Space

A message typically provides time-stamp information in its header. This information is parsed to extract the hours, minutes and seconds from the time-stamp (see figure 4.15 for an example message).

```
Date: Wed, 1 Apr 98 21:20:49 EST
From: Operator <root@media.mit.edu>
Subject: Voice message from 253-4086
```

Figure 4.15: Extracting the spatial angle from the time-stamp information obtained from a typical voice message.

The spatial angle of the audio source is then computed, based on its time of arrival and the predefined mode of display (12 or 24-hour):

$$\text{Spatial Angle in degrees } \theta_t = [\ 90° - (\ (\text{Message}_{hrs} \text{ x } (360°/24 \text{ hrs}) \text{ x } \text{Clock}_{scale}) +$$

$$(\text{Message}_{mins} \text{ x } (360°/(24 \text{ hours x } 60 \text{ mins})) \text{ x } \text{Clock}_{scale}) +$$

$$(\text{Message}_{secs} \text{ x } (360°/(24 \text{ hours x } 3600 \text{ secs})) \text{ x } \text{Clock}_{scale})\ )\ \ ]$$

where Clock_{scale} = 1 if display mode = 24-hour or Clock_{scale} = 2 if display mode = 12-hour and 90° is the initial position on the spatial display i.e. 12:00 AM

The spatial angle computed (θ_t degrees) is converted from Angular to Cartesian coordinates to position the message on an x-y-z plane.

$$\text{Spatial Angle in radians } \theta_r = (\ (\ \theta_t\ ) \text{ x } \pi/ 180\ )$$

$$X_{position} = (\ X_{size}/2 + (\ \text{distance x Cosine } (\ \theta_r\ )\ )\ )$$

$$Y_{position} = (\ Y_{size}/2 + (\ \text{distance x Sine } (\ \theta_r\ )\ )\ )$$

The distance here is computed based on the category of the sound source. Voice messages are placed closer than email or news summaries. The distance changes when the sound must be pulled closer to the listener during scanning and is used to re-compute the spatial position dynamically.

The elevation of the sound $Z_{position}$ = 0 if the message must be localized in the foreground (on the listener's plane of hearing) or, it is set higher to $Z_{position}$ = 2.5 if the message must be localized in the background. This effectively controls the relative gain level of the sound, especially since its (x, y) position is predefined by its time of arrival.

Spatial location is then rendered by this function:

$$\text{PositionSound } (X_{position,}\ Y_{position,}\ Z_{position})$$

The listener is positioned at the center of the audio space and zero elevation:

$$\text{PositionListener}(X_{size}/2,\ Y_{size}/2,\ 0)$$

4.7.6 Modes of Spatial Listening

Spatial listening can be utilized for several different modes of message delivery based on the need to present a single audio message or a browse/scan through a sequence of messages:

Broadcasting

When new messages arrive, they are *broadcast* to the listener from a specific spatial location. These messages are heard in the *background* and fade away if the user does not explicitly activate message playback. This mode is based on the metaphor of traditional radio broadcasting where listeners passively listen to news stories and only pay attention when a relevant article is heard.

Browsing and Foregrounding

Browsing is an active form of listening where users can select a category and browse sequentially through all messages, playing each one as needed. This mode is similar to the metaphor of switching stations on a radio until a station playing desirable music is found. When a desirable message is heard, the user can stop and listen to the entire message in the *foreground*. The user can specify the duration of the message by selecting one of three playback scales: summary (2.5 seconds), preview (1/5th of the message) or the full message. The audio source of the message is moved closer to the listener from its current location within a period of 1 second, and played there for 4/5th of the user-defined duration (see figure 4.16). Finally, the message gradually begins to fade away for the remaining 1/5th of the duration, returning to its original location. The *foregrounding* algorithm ensures that the messages are quickly brought into perceptual focus by pulling them to the listener rapidly. Yet the messages are pushed back slowly to provide an easy fading effect as the next one is heard. As the message is pulled in, its spatial direction is maintained allowing the listener to retain a sense of message arrival time. This spatial continuity is important for discriminating and holding the auditory streams together [Arons92].

Spatial Scanning

Sometimes listeners want to get a preview of all their messages quickly without manually selecting and playing each one. This is similar to the *scan* feature on modern radio tuners that allows users to alternatively hear each station for a short duration. In *Nomadic Radio*, message scanning cycles through all messages by moving each one to the center of the listening space for a short duration of time and fading it out as the next one starts to play (see figure 4.17). All messages are played sequentially in this manner, with some graceful overlap as one message fades away and the next one begins to play. The *scanning algorithm* interlaces audio streams in parallel by running each one in its own thread. This simultaneity allows the listener to hear an overall preview of his message space in an efficient manner with minimum interaction.

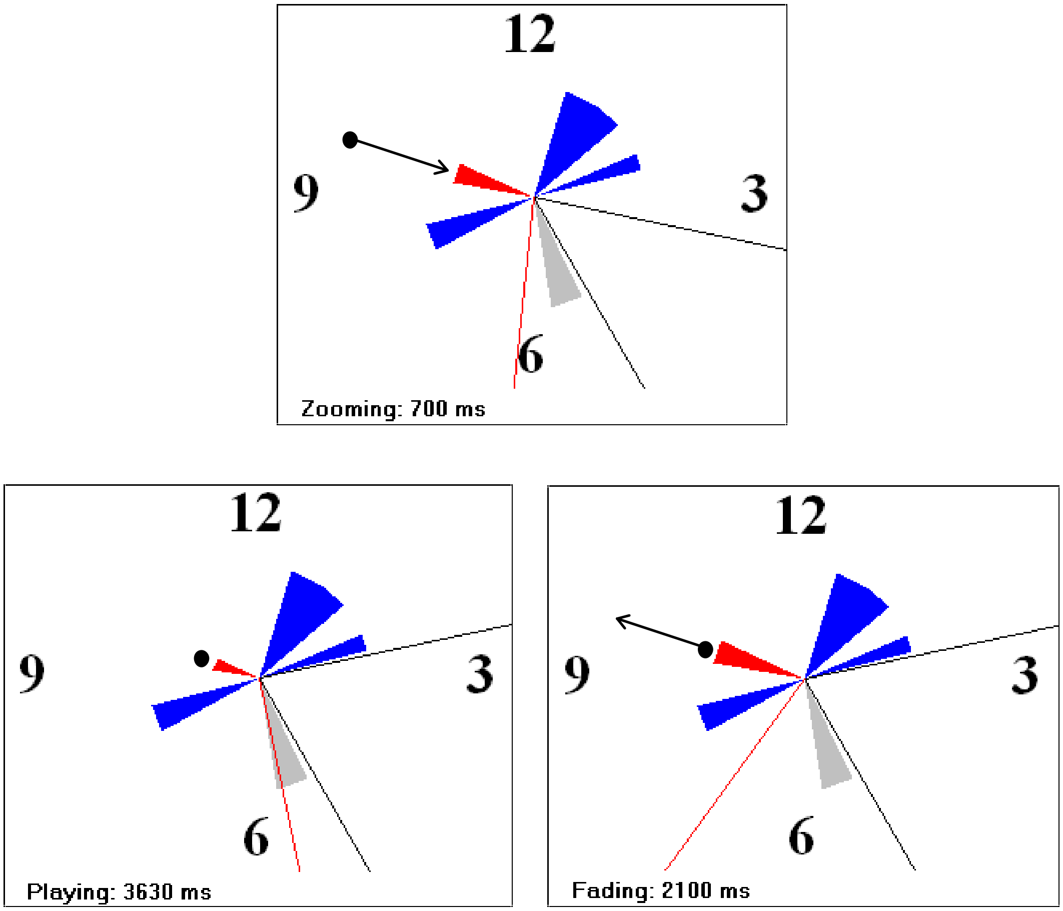


Figure 4.16: *Foregrounding* an audio message by quickly pulling it closer to the listener (zooming) from its temporal location in the audio space. The message plays there for 4/5th of its duration before gradually fading back to its original location.

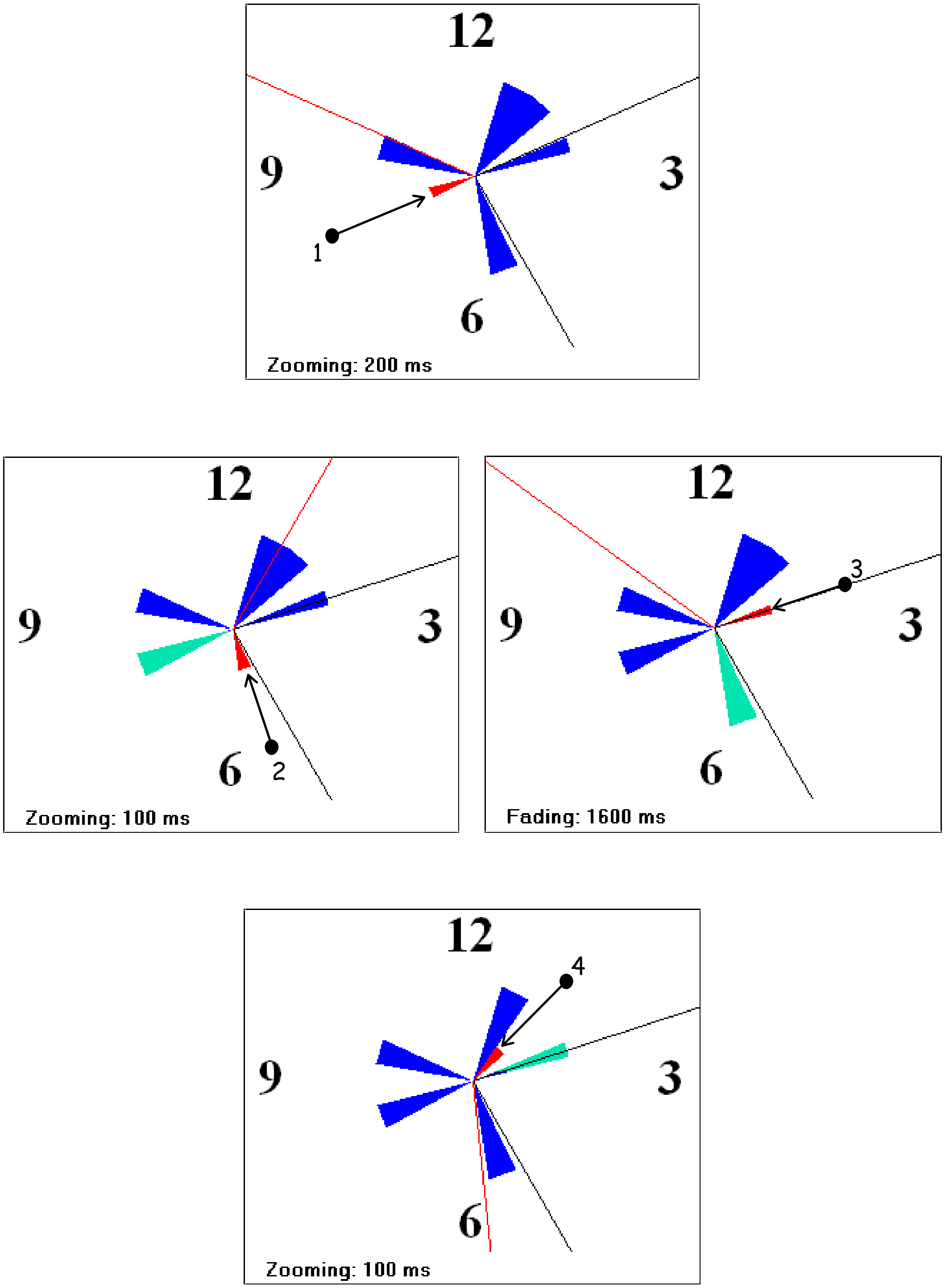


Figure 4.17: Scanning a sequence of voice messages (1-4) where as one fades out, the next one zooms in, overlapping with the previous message for a few seconds.

Previewing via Time-Compressed Foregrounding

Audio streams can be previewed more efficiently by playing their content faster to the listener, although this technique reduces comprehension. In the current version of *Nomadic Radio*, a simple form of time-compression is implemented by simply shifting the pitch rate of the audio and playing it up to 1.3 times its original playback rate (better techniques are discussed next). In this implementation, a message is previewed by using a combination of time-compression and spatial audio. The message is initially played from a specific spatial location around the listener and the pitch rate is gradually increased from 1.0 to 1.3x speed while the message is brought within spatial proximity of the listener (see figure 4.18). As the message starts to fade-back the pitch rate is again gradually decreased to its original rate of playback. This allows the listener sufficient time to recognize the speaker of the message and get accustomed to the characteristics of the compressed voice. It should be noted that this is an experimental technique and has not been readily evaluated with listeners.

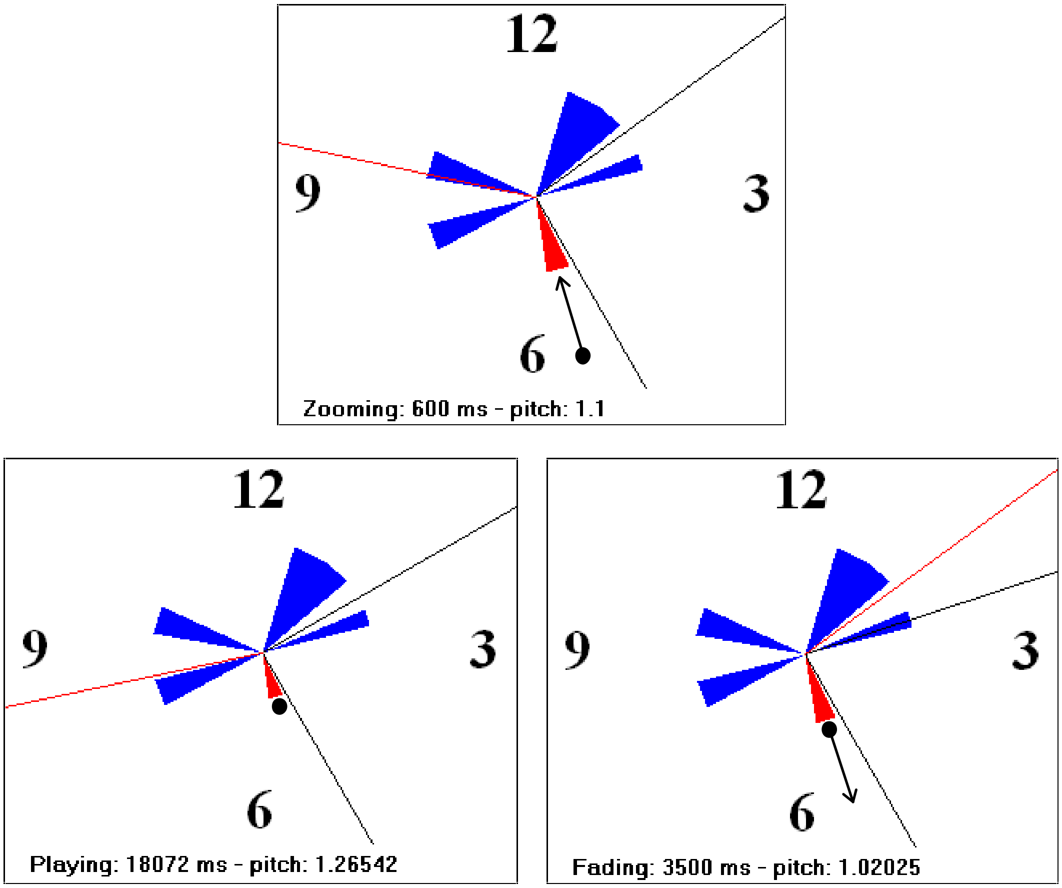

12
9
3
6
Zooming: 600 ms - pitch: 1.1
12
9
3
6
Playing: 18072 ms - pitch: 1.26542
12
9
3
6
Fading: 3500 ms - pitch: 1.02025

Figure 4.18: Message previews via spatial time-compression. The pitch of the sound increases as it zooms, reaching 1.3x speed before slowing back down as it fades out.

Time compression was incorporated in *AudioStreamer* [Mullins96] as an experiment where a selected stream could be played gradually up to 2x times faster until the end of audio was reached. AudioStreamer did not foreground the audio stream during time-compression, and there was no gradual slow down to the original playback rate.

Time compression by increasing the pitch rate creates a frequency shift in the sound leading a decrease in intelligibility (distorting the speaker's voice). In *Nomadic Radio*, a maximum pitch of 1.3x times was used to maintain intelligibility. There are a range of techniques for time-compressing speech without modifying the pitch, based on SOLA (synchronized overlap add method) [Arons92]. Research suggests that presenting the audio at over twice its original playback rate, provides too little of the signal to be accurately comprehensible [Heiman86]. Comprehension of time-compressed speech does increase as listeners hear it more frequently since they tend to adapt to it quickly [Voor65]. Novice users quickly adapt to a compression rate of 50%. But with more training, much higher compression is possible. In future versions of *Nomadic Radio*, SOLA-based compression will provide more effective time compression for previewing messages.

It is interesting to consider how time-compression used in conjunction with spatial and simultaneous listening can enhance browsing efficiency. A controlled experiment can ascertain the effect of this technique. For certain types of content such as news broadcasts, listeners may be more inclined to hear them as time-compressed streams in the background while a timely voice message plays at normal speed in the foreground. When a relevant event is heard in the news stream, the listener may be able to shift auditory focus back to it and play it in the foreground at its normal speed.

We have described a number of techniques for browsing, scanning and previewing spatial audio messages, using a non-visual display. Yet, we have found that demonstration of these spatial audio effects inherently requires a visual language shown in the figures earlier. We will now consider the design of the visual representation and discuss the reasons for providing an auxiliary visual display.

4.8 Auxiliary Visual Display

In *Nomadic Radio*, several visual interface components were created over time, primarily for development purposes. While the messaging infrastructure was being developed, a mechanism was needed for displaying the messages received and browsing them easily. Therefore, interface components for messaging, i.e. the *Message Window* and *Browsing Control*, were created (see components 3 and 4 in figure 4.20). As new categories of messages were added, the *Radio Channel Control* (component 2) permitted easy switching between views and modes of operation (visual, sleep and graph). Additional components were added to test functionality such as audio playback modes and graph parameter controls.

As spatial audio functionality was added to the system, a visual display was needed for visualization of the localization techniques developed for *Nomadic Radio*. The display allowed rapid prototyping and experimentation with different spatial audio layouts and browsing/scanning techniques. Not surprisingly, the display also permitted an effective way of explaining the functionality and spatial layout design, during demonstrations to others. The initial display created, positioned the listener at the center (see cross-hair in figure 4.19) and allowed the user to move the listener's position around the audio space. On a wearable this was found to be an impractical design and the listener position was fixed to the center. This display visually represented messages in a somewhat cluttered and less intuitive manner. As a result a simplified visual representation, the *Message Swatch*, was adopted (discussed further in the next section).

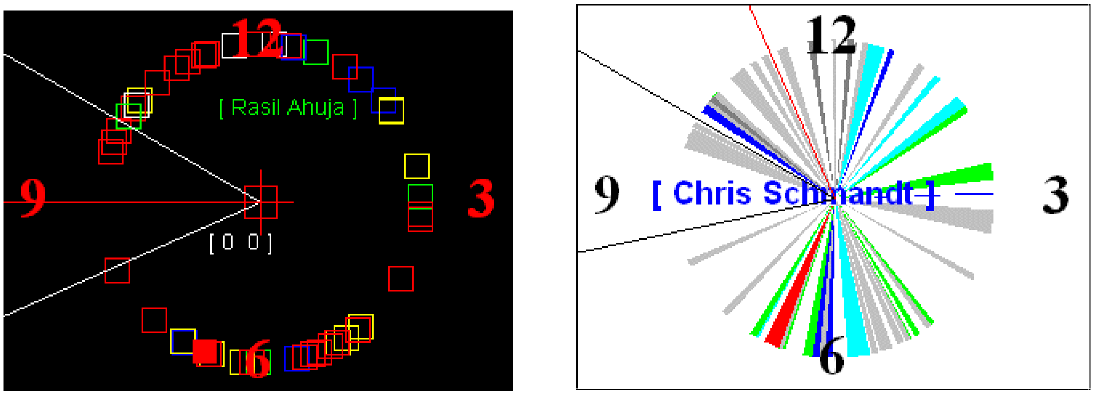
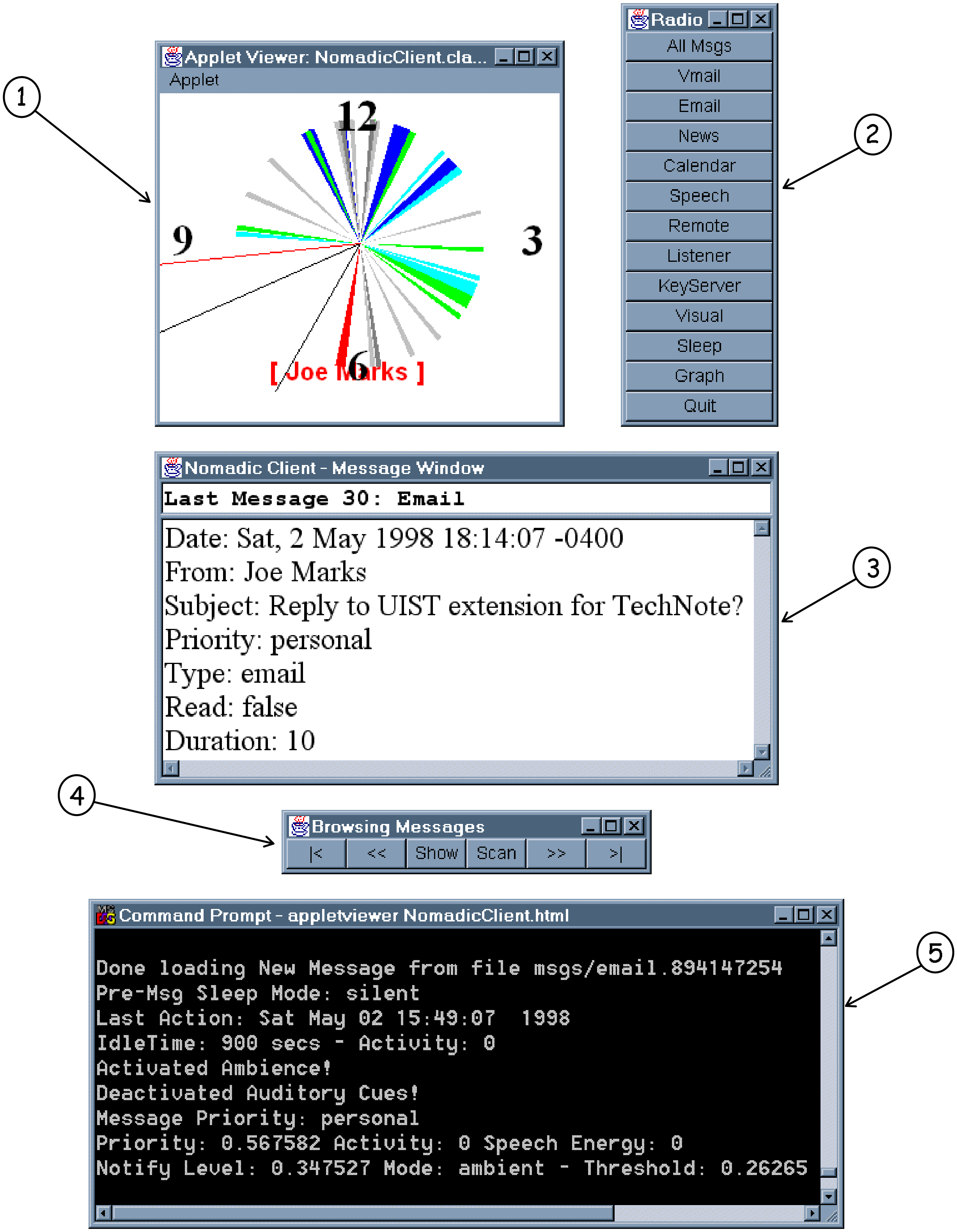


Figure 4.19: Iterative design of the visual display for visualization of spatial layout.

Overall, the display allows one to see a summary of all messages at a glance and permits users to observe the relative notification levels for messages over time (see interactive notification graph in chapter 6). The visual interface components are necessary during setup of the system, to change operational parameters, and observe aspects of the system during its usage (see *Info. Console* in Figure 4.20). An effective display can augment the auditory functionality of *Nomadic Radio,* if the display is unobtrusive. An auditory space complemented with a simple visual interface on a small wrist-worn device (similar to a watch) can provide bi-modal interaction and visualization in several situations.



Last Message 30: Email

Date: Sat, 2 May 1998 18:14:07 -0400
From: Joe Marks
Subject: Reply to UIST extension for TechNote?
Priority: personal
Type: email
Read: false
Duration: 10

```
Done loading New Message from file msgs/email.894147254
Pre-Msg Sleep Mode: silent
Last Action: Sat May 02 15:49:07  1998
IdleTime: 900 secs - Activity: 0
Activated Ambience!
Deactivated Auditory Cues!
Message Priority: personal
Priority: 0.567582 Activity: 0 Speech Energy: 0
Notify Level: 0.347527 Mode: ambient - Threshold: 0.26265
```

Figure 4.20: The visual interface components in *Nomadic Radio*: (1) Message Swatch, (2) Radio Channel Control, (3) Message Window, (4) Browsing Control, (5) Info. Console

4.8.1 Design of Message Swatch

The metaphor of a watch is used to create an interface to represent dynamic personal information. The *Message Swatch* has a simple visual display which allows users to understand its chronological representation at a glance. The spatial display of the watch dial has additive affordances [Norman94] which allows representation of different message attributes using geometric and color attributes of the display (see figure 4.21). Messages are displayed based on their time of arrival, importance and duration. Concentric lines overlaid on the dial represent messages with varying attributes such as line thickness and brightness. Thickness indicates relative duration of the message whereas color and brightness indicate the inferred priority of the message. Here voice messages are shown in the inner ring as smaller lines and email is shown as longer lines. A greater number of incoming email messages, requires a larger space for display in a radial form. The current message is highlighted as red, and pop-up text shows the sender when a cursor is moved over the message. Scheduled events are shown as shaded areas of the dial, dynamically changing (in brightness) as the event gets closer in time.

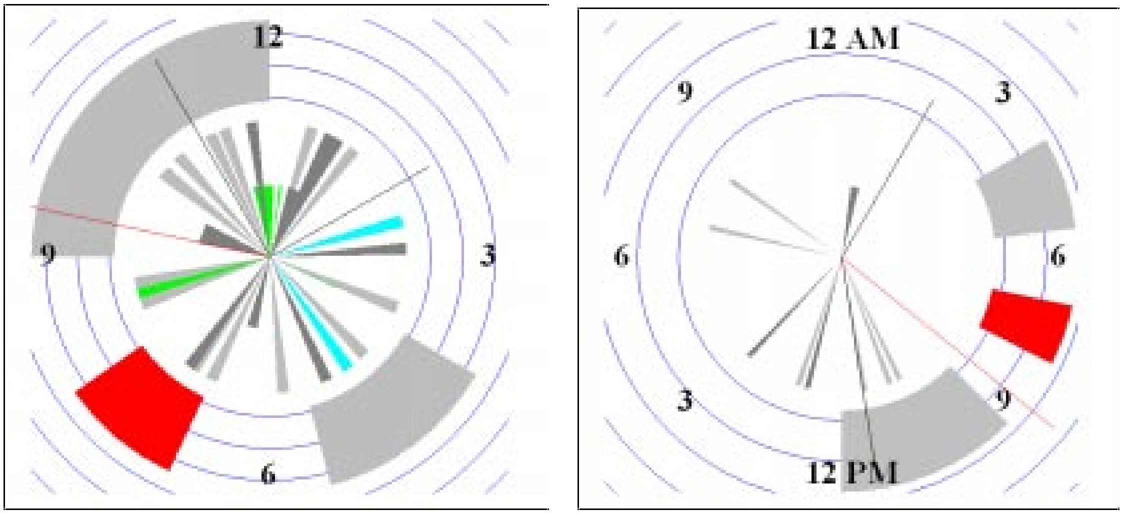


Figure 4.21: Display of information in Message Swatch using a 12 and 24-hour format. Voice messages are shown in the inner ring and email messages as longer lines. The shaded areas represent calendar events, with the current one highlighted.

The scalability of the display is an issue as the number of messages increase. The user filters the messages shown on the display by selectively requesting a category, unread messages or messages for the current day. This helps reduce the number of messages shown concurrently and makes the information space more manageable for an overview of relevant messages. The size of a message region reflects its length or duration, yet this is problematic for messages of widely varying sizes (5 to 5000 lines of content). This issue is resolved by using logarithmic scaling of the size to display the messages of varying duration in a manageable form. A continuously varying representation of messages as concentric lines, creates an implicitly aesthetic design for a personal interface on a watch or auxiliary visual display.

5. Scaleable and Contextual Notification

Nomadic Listeners value their time and don't wish to be interrupted by continuous audio broadcasts and recurring notifications throughout the day. This chapter discusses problems with notifications on current portable devices that cause undesirable interruption to listeners and inhibit their usage in social environments. An extended usage scenario describes the audio notification techniques used in *Nomadic Radio* based on a model of the user's attentiveness. Key aspects of this model include *scaleable notification*, *contextual cues* and *dynamic operational modes*. Positive and negative reinforcement from the listener allows the system to continuously adjust its notification model over time. Evaluation of preliminary data from actual usage in *Nomadic Radio* demonstrates some indication of effectiveness of this model and the auditory and visual interaction techniques developed.

5.1 The Nature of Interruptions in the Workplace

Users organize their work during certain intervals of the day, which may be considered their "timespace in the workplace". A recent observational study [Conaill95] evaluated the effect of interruptions on the activity of mobile professionals in their workplace. The study analyzed 29 hours of video data to extract 125 naturally occurring interruptions. An interruption, defined as asynchronous and unscheduled interactions, not initiated by the user, results in the recipient discontinuing the current activity. The results revealed several key issues:

- On average, subjects were interrupted over 4 times per hour, for an average duration slightly over 2 minutes. Nearly 10 minutes per hour were wasted on interruptions.
- A majority of the interruptions occurred in a face-to-face setting. Yet 20% (25 interruptions) were due to telephone calls (no email or pager activity was analyzed in this study).
- In 64% of the interruptions, the recipient received some benefit from the interaction.
- In 41% of the interruptions, the recipients did not resume the work they were doing prior to the interruption.

This suggests that a blanket approach to prevent interruptions, such as holding all calls or face-to-face interactions at certain times of the day, would prevent beneficial interactions from occurring. Active use of new communication technologies makes users easily vulnerable to undesirable interruptions. These interruptions constitute a significant problem for mobile professionals using tools such as pagers, cell-phones and PDAs, by disrupting their time-critical activities. Improved synchronous access using these tools benefits initiators but leaves recipients with little control over the receipt of the interactions, e.g. undesirable phone calls or notifications from pagers in inappropriate environments.

The study suggests development of improved filtering techniques that are especially light weight, i.e. don’t require more attention on part of the user or the filtering process be just as disruptive as the interruption itself. By moving interruptions to asynchronous media, messages can be stored for retrieval or notification at a more appropriate time. In *Nomadic Radio,* the use of auditory cues provides light-weight notification. The messages are delayed and presented on the basis of the user’s inferred interruption level.

5.2 Notifications in Everyday Environments

In today’s environment, people use a number of appliances and portable devices for a variety of tasks in the home, workplace and on the run. Such devices are ubiquitous and each plays a unique functional role in a user’s lifestyle. Each device requires some form of active or passive interaction with users via tactile feedback as well as auditory and visual displays. To be effective, these devices need to notify users of changes in their functional state, incoming messages or exceptional conditions. In a typical office environment, the user attends to a plethora of devices with notifications such as calls on telephones, asynchronous messages on pagers, email notification on desktop computers, and reminders on personal organizers or watches (see figure 5.1). Such an environment poses a number of key problems discussed below.

5.2.1 Lack of Differentiation in Notification Cues

Every device provides some unique form of notification. In many cases, these are distinct auditory cues. Yet, most cues are generally *binary* in nature, i.e. they only convey the occurrence of a notification and not its urgency or dynamic state. This prevents the user from making timely decisions about the messages received without having to shift their focus of attention (from their primary task) to interact with the device and access the relevant information.

5.2.2 Lack of Unified Notifications

All devices compete for a user’s undivided attention without any coordination and synchronization of their notifications. If two or more notifications occur within a short time of each other, the user gets confused or frustrated. As people start carrying around many such portable devices, frequent and uncoordinated interruptions inhibit their daily tasks and interactions in social environments.

5.2.3 Lack of Situational Awareness of the User and Environment

Such notifications occur without any regard to the user’s engagement in her current activity or her focus of attention. This interrupts a conversation or causes an annoying disruption in the user's task and flow of thoughts. To prevent undue embarrassment in social environments, users typically turn off cell-phones and pagers in meetings or lectures. This prevents the user from getting notification of timely messages and frustrates people trying to get in touch with the user.

5.2.4 Lack of Adaptive Behavior to Learn from Prior Interactions

Such systems typically have no mechanism to adapt their behavior based on the positive or negative actions of the user. Pagers continue to buzz and cell-phones do not stop ringing despite the fact that the user may be in a conversation and ignoring the device for some time.

Given these problems, most devices fail to serve their intended purpose of notification or communication, and thus do not operate in an efficient manner for a majority of their life-cycle. New users choose not to adopt such technologies, having observed the obvious problems encountered with their usage. In addition, current users tend to turn off the devices in many situations, inhibiting the optimal operation of such personal devices.

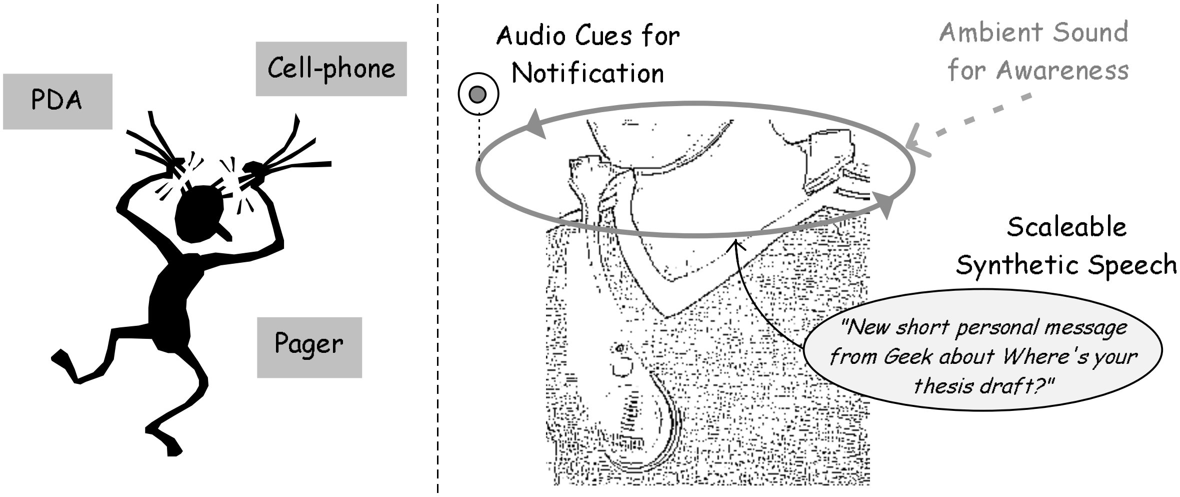

PDA
Cell-phone
Pager
Audio Cues for Notification
Ambient Sound for Awareness
Scaleable Synthetic Speech
"New short personal message from Geek about Where's your thesis draft?"

Figure 5.1: Everyday personal devices vying for the user's attention at different times of day vs. the use of unified and scaleable notification in *Nomadic Radio.*

The problem of effective notification is especially relevant in critical situations where several aspects of the environment must be continuously monitored while maintaining communication with coworkers. Such environments include hospital emergency rooms, satellite mission control, warnings in aircraft cockpits, process control in production plants, and even network administration. In these situations, time critical information can be delivered to health-care specialists (doctors and nurses), operators and administrators, without overwhelming them with redundant data while effectively managing their focus of attention between communication and monitoring tasks. A notification model is evaluated by considering how well it maximizes the value of information delivered and minimizes the cost of disrupting the user.

A scenario for personal messaging and communication (demonstrated in *Nomadic Radio*) provides a simple and constrained problem domain in which to develop and evaluate an adaptive notification model. This scenario requires development of a model that continuously learns a suitable *notification strategy* based on usage patterns of a specific user in certain environmental contexts. Such a system could infer the user's attention level by monitoring her current activities such as interactions with the device and conversations in the room. The user's prior responses to notifications must also be taken into consideration to adapt the notifications over time. We will consider scaleable notification techniques and an appropriate parameterized approach towards *contextual notification* in *Nomadic Radio*.

5.3 Approaches for Managing Attention in Nomadic Radio

5.3.1 Active Browsing and Passive Listening

We have discussed active and passive modes of listening where information is delivered as synthetic speech and spatialized audio. An active listener primarily selects categories and *browses* the messages sequentially using a combination of speech and tactile interaction. Alternatively, the listener can activate *scanning* to quickly listen to a sequence of message summaries before selecting a relevant message for playback. A technique such as *scanning* represents a hybrid form of access, where passive listening is coupled with automatic browsing. Finally, a somewhat more passive form of listening requires the information to be *continuously broadcast* to the listener at predefined intervals (for news summaries and calendar reminders) or broadcast on a timely basis (for email and voice messages). The notion of *broadcasting*, which originated in the medium of Radio, has been adopted in recent web-browsers (such as *PointCast* and *MS Internet Explorer 4.0*) providing updated Internet content via a number of user-defined *channels*. Such an "information-push" model of content delivery is only as effective as the relevance of the information to the listener based on their current context and their ability to attend to it given the nature of their tasks. *Scaleable broadcasting* coupled with *contextual and timely notification* of prioritized information allows a listener to selectively attend to pertinent content.

5.3.2 Unified Notification Strategy

If notifications are coordinated, then they would be queued appropriately such that the important ones are delivered to the user first and other notifications even aborted. Alternatively, such individual notifications are fused and summarized as an abstract representation that conveys this information in a condensed and usable manner to the listener.

5.3.3 Distinct Notification Cues

A well designed collection of coherent auditory cues can convey complex states and subtle changes in notifications and provides the listener with a higher level of awareness in less time. The user makes appropriate decisions about how to best handle the situation, without expending additional effort to interact with the device and access related information. This approach implicitly reduces transaction time and cognitive overhead in using such devices over long periods of time.

5.3.4 Pre-cueing via Notification Latency

Pre-cueing is a phenomenon that cognitively prepares listeners to open a channel of attention [Pashler98], based on a prior warning of an event. In the contextual notification model developed for *Nomadic Radio,* auditory cues are pre-cue the listener's attention before delivering incoming messages.

5.3.5 Situational Awareness

Continuous awareness of the user and environment enables devices to scale down their notifications when the user is inferred to be in a conversation or busy with other tasks. If the user needs to be notified, the device provides an appropriate notification when the user seems most interruptible.

5.3.6 Adaptive Notification

Several machine learning techniques have been used in the past to control display of information or 3D graphics renderings, based on the resources available, perceptual cost of degradation and attentional focus of viewers [Horvitz 95, 97]. By monitoring usage and detecting a user's level of satisfaction with notification presented, personal devices can upgrade or degrade their notification strategy. This process enables the system to learn when it is most appropriate to interrupt the user.

5.4 An Extended Scenario for Notification

Before discussing details of the model, lets us first consider an example scenario. This scenario demonstrates how contextual cues affect the system's operational modes and scale notifications over time. The scenario focuses on the dynamics of the audio presentation and not speech input.

Jane uses *Nomadic Radio* frequently to access her personal messages and receive calendar reminders, when she is away from her office.

It's 1:15 PM and Jane is wearing *Nomadic Radio*. She has a meeting in a conference room in 15 minutes. The system gives her an early notification about the meeting via an auditory cue and synthetic speech.

NR: <auditory cue for early event reminder> ***"Jane, you have a scheduled event at 1:30 PM today."*** **<pause>** ***"Meeting with Motorola sponsors in NIF conference room for 30 minutes."***

Jane scans her email messages to hear one about the meeting and check who else is coming. The system is currently in *active speech* mode.

Jane walks over to the kitchen to get some coffee and stops to speak with a colleague who meets her in the hallway. Moments later she hears an auditory cue notifying her of an incoming group message (a low priority message).

NR: <auditory cue for group message>

The system notices that Jane was recently browsing her messages, and that there is a lower level of conversation in the room. After the auditory cue, Jane hears the ambient sound of water speed up gradually for 7 seconds and realizes that she is about to hear a message. Jane sips her coffee and takes no action. The system then plays a summary of the message.

NR: <ambient sound speedup to fast rate and slows down> ***"New short group message from Chris Schmandt about 'Has anyone seen my glasses?'"***

Jane ignores the message and pours more coffee in her cup.

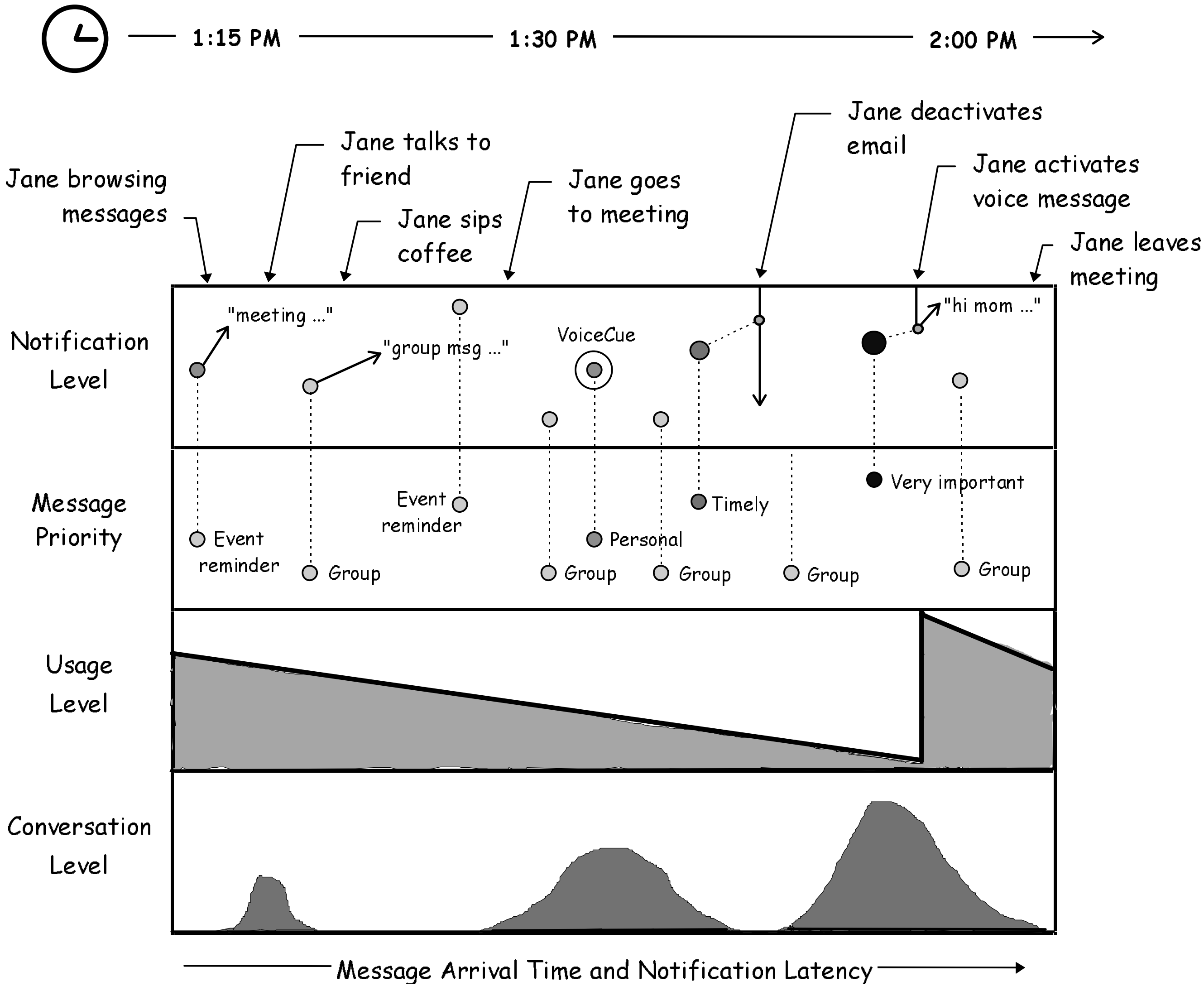


Figure 5.2: An illustration of notification levels and latency computed for messages with different priorities, under varying levels of usage and conversation.

The system notices that Jane has not been actively using it for over 5 minutes, so it turns off all speech feedback and enters *auditory mode.* Three minutes before the meeting, *Nomadic Radio* gives Jane another reminder, this time with a subtle audio cue only and no spoken feedback (since she heard it earlier).

NR: <audio cue for timely event reminder>

Jane can press a button to hear the text of the meeting reminder. Jane heads over to the conference room. At this point, since Jane has been inactive for some time and the conversation level in the room is higher, the system scales down notifications for all incoming group messages. A few minutes into the meeting, a personal message arrives and the system notices a high level of conversation in the room. It now plays auditory cues to notify Jane.

NR: <auditory cue for personal message> + <VoiceCue identifying the sender>

Jane recognizes the voice of her friend Susan, and knows that she can get back to her message later and continues participating in the meeting. Moments later a timely message arrives (related to an email Jane sent earlier in the day) and the conversation level is somewhat lower, so the system first plays an audio cue and gradually speeds up the sound of water to indicate to Jane that she will hear a summary shortly after.

NR: <auditory cue for timely message> + <ambient sound speedup>

Jane is now engrossed in the meeting so she prevents the system from playing a summary of the message, by pressing a button on *Nomadic Radio* (she does not speak to avoid interrupting the meeting). The sound of water slows down and the message is not played. The system recognizes Jane is busy and turns down the level of all future notifications.

Its 1:55 PM and the meeting is nearly over. The system, is currently in *sleep* mode. A very important voice message from Jane's daughter arrives. The system switches to *active speech* mode. It recognizes the priority of the message and despite the high conversation level and low usage level, it plays audio cues to notify Jane. The ambient sound is speed-up briefly to begin playing a preview of the message in 3.5 seconds.

NR: <audio cue for voice message "telephone ringing" sound> + <VoiceCue of Jane's daughter>

Jane hears her daughter's voice and immediately presses a button to play the message. The system starts playing the full voice message in the foreground (instead of just a preview), 2 seconds earlier than its computed latency time.

NR: <human voice> ***"Hi mom, its Kathy. Can you pick me up early from school today? .... oh and don't forget to bring my ballet shoes too".*** **<audio cue indicating end of voice message>**

> Jane excuses herself from the meeting and remembers that she needs to email Susan before leaving work to get Kathy. She browses Susan's email on *Nomadic Radio* while walking back to her office to get her car keys.

As this extended scenario demonstrates, notifications are scaled appropriately to minimize interruptions when the user is busy. It also shows how the user can actively influence the continuous notification model by her positive or negating actions that reinforce or scale down future notifications. This model will be described in detail in section 5.8. Such a model requires three key aspects: *Scaleable Notification*, *Contextual Cues* and *Dynamic Operational Modes*. We now examine how these aspects interact to actively determine an effective notification strategy for the user.

5.5 Scaleable Auditory Presentation

In *Nomadic Radio*, several sources of information are broadcast to the listener, described as *active information services* in section 3.2. A unified audio interface allows access to all services in a familiar and consistent manner. The contents of prioritized email and calendar events are presented as synthesized speech along with auditory cues to convey the priority of the message and identification of the sender (*VoiceCues* for email). Voice messages and ABC News summaries are rendered as spatial audio sound sources based on their time of arrival (discussed in section 4.7.5). A common set of voice commands treats all message types in the same manner, i.e. by saying "move forward" or "read this message". Similarly, messages are scaled and presented consistently for different modes of access such as browsing, summarizing and notifications. A scaleable presentation is necessary for delivering sufficient information while minimizing interruption to the listener.

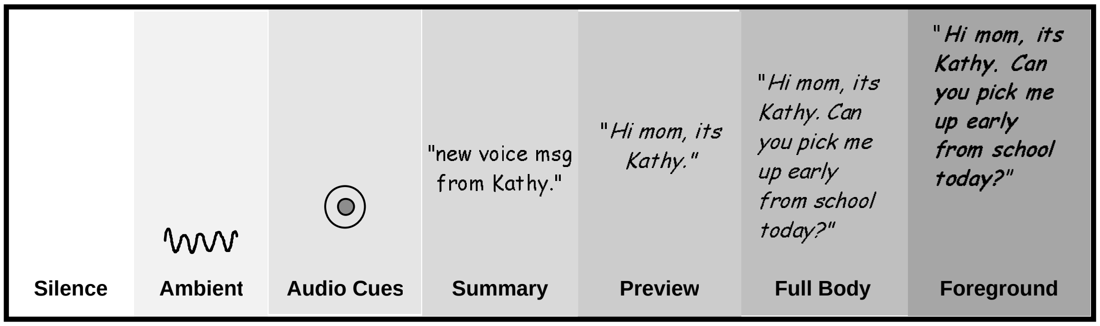

"new voice msg from Kathy."
"Hi mom, its Kathy."
"Hi mom, its Kathy. Can you pick me up early from school today?"
"Hi mom, its Kathy. Can you pick me up early from school today?"
Silence
Ambient
Audio Cues
Summary
Preview
Full Body
Foreground

Figure 5.3: Dynamic scaling of an incoming voice message during its life-cycle, based on the interruptability of the listener. The message is presented at varying levels: silent notification, to playing a subtle audio cue, to increasing levels of audio presentation.

Messages in *Nomadic Radio* are scaled dynamically to unfold with increasing levels of notification, described below:

5.5.1 Silence for Least Interruption and Conservation

When a user is busy or simply not available to listen to messages, no feedback about an incoming message is preferable to the system continuously playing audio cues and speaking the message content. In this mode all auditory cues and speech feedback is turned-off. Messages can be scaled down to silence (scale-zero) when the message priority is inferred to be too low for the message to be relevant for playback or awareness to a user, based on her recent usage of the device and conversation level (these act as parameters for a contextual notification model, developed in the next section). This mode also serves to conserve processing, power and memory resources on a portable device or wearable computer.

5.5.2 Ambient Audio for Peripheral Awareness

A listener is aware of an incoming message and its size via changes in the continuously playing ambient audio stream. By listening to the ambient audio stream, the listener can gauge the size and type of incoming message. In *Nomadic Radio*, short email messages are heard as "splashes" in the ambient flowing water, whereas longer news summaries increase the pitch of the ambient stream (heard as water flowing much faster). Such a *peripheral awareness* minimizes cognitive overhead of monitoring incoming messages relative to notifications played as distinct auditory cues which incur a somewhat higher cost of attention on part of the listener.

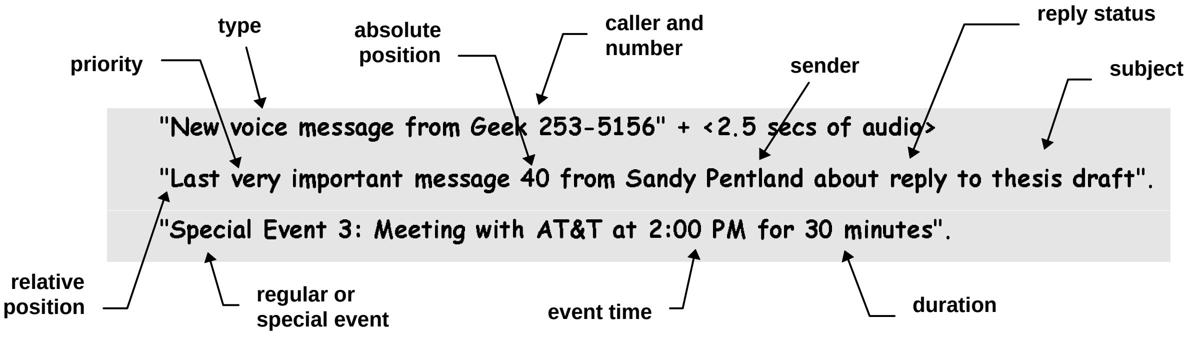
5.5.3 Auditory Cues for Notification and Identification

A compact representation of a message serves as an effective cue to determine the priority of a message and the identity of the sender. In *Nomadic Radio*, when a new message is downloaded, the user is notified of an incoming message via message priority cues and *VoiceCues*. While browsing or scanning the messages, these auditory cues are played to identify the messages in a non-visual environment. A classification of auditory cues used in *Nomadic Radio* is described in section 4.6. Once a listener is familiar with the parameters associated with an auditory cue, it provides an efficient representation that is less distracting than a corresponding textual description rendered via synthetic speech feedback.

5.5.4 Generating a Message Summary

A spoken description of an incoming message can present the relevant information in a concise manner. Such a description typically utilizes header information in email messages to convey the name of the sender and the subject of the message. In *Nomadic Radio*, message summaries are generated for all messages, including voice-mail, news and calendar events. The summaries are augmented by additional attributes of the message (see figure 5.4) indicating category, order (first, last, new, or message number), priority (personal, group, very important and timely) and duration (short, long, very long). The design of the spoken dialogue for composing such summaries and related examples are in section 4.5. For audio sources, like voice messages and news broadcasts, the system also plays the first 2.5 seconds of the audio.

This identifies the caller and the urgency of the call, inferred from the intonation in the caller’s voice. It also provides a station identifier for news summaries.



"New voice message from Geek 253-5156" + <2.5 secs of audio>

"Last very important message 40 from Sandy Pentland about reply to thesis draft".

"Special Event 3: Meeting with AT&T at 2:00 PM for 30 minutes".

Figure 5.4: Message summaries with the important attributes that are spoken.

5.5.5 Content Summarization for Message Previews

Messages are scaled to allow listeners to preview the contents of an email or voice message in a shorter time duration. In *Nomadic Radio*, a preview for text messages extracts the first 100 characters of the message (a default size which can be user defined). This heuristic generally provides sufficient context for the listener to anticipate the overall message theme and urgency. For email messages, redundant headers and previous replies are eliminated from the preview for effective extraction. Use of text summarization techniques, based on tools such as *ProSum*[17] developed by British Telecom, would allow more flexible means of scaling message content. Natural language parsing techniques used in *ProSum* permit a scaleable summary of an arbitrarily large text document.

A preview for an audio source such as a voice message or news broadcast plays a fifth of the message in the foreground (described in section 4.7.6). A better representation requires a structural description of the audio, based on annotated or automatically determined pauses in speech, speaker changes [Roy95] and topic changes [Stifleman97]. Such a representation is considered an *auditory thumbnail* similar in function to its visual counterpart. A preview for a structured voice message allows a listener to hear pertinent aspects such as name of caller and phone number. ON the other hand, a preview for a structured news broadcast would be heard as the prominent headlines for the day.

Another mechanism for previewing the contents of text and audio messages is increasing its rate of playback. For text messages, this is accomplished via changing the "speaking rate" of the speech synthesis module (AT&T’s *FlexTalk Engine*). Listening to synthetic speech already requires greater cognitive overhead, and when it is sped-up, it is nearly incomprehensible for most listeners. As a result, text summarization is a preferred technique. For audio-based messages there are a range of techniques for

[17] *http://transend.labs.bt.com/prosum/on_line/*

time-compressing speech without modifying the pitch, yet twice the playback rate makes the audio incomprehensible.

5.5.6 Reading or Playing Complete Message Content

This mode plays the entire audio file or reads the full text of the message at the original playback rate. Some parsing of the text is necessary to eliminate redundant header information and format tags. The message is augmented with summary information indicating sender and subject. This message is generally played in the background of the listener's audio space.

5.5.7 Spatial Proximity for Foregrounding Messages

An important message is played in the foreground of the listening space. For an audio message the spatial position determines time of arrival. A gradual fading-in effect (as described earlier in section 4.7.6) eases listening to the message. Finally, towards the end of the message, the audio gradually fades out indicating message completion to the listener. For text messages, synthesized via synthetic speech, the volume of spoken text increases to foreground the message. Other features of the synthetic voice such as pitch (changes mood and expressiveness of the voice) and gender of the synthetic speaker [Cahn90] can be used in the future to indicate higher priority messages.

The scale of auditory presentation is set while browsing by manually activating/deactivating speech feedback, auditory cues or the ambient sound. The user selects a playback mode as: "play {summary | preview | full message | foreground }" via spoken commands or tactile interaction. In the following sections, we will discuss two implicit mechanisms for automatically scaling message presentation in *Nomadic Radio*, based on the system's *dynamic operational modes* and a *contextual notification model*. In addition to auditory scaling, we will consider the issue of *latency* in incoming message delivery as a technique for *pre-cueing* the listener's attention. Automatic scaling requires inferring the urgency of the message, activity of the user and attributes of the environment. Such contextual sensing permits the system to adjust its presentation level to minimize interruption while delivering appropriate level of information content to the listener.

5.6 Sensing and Analyzing Contextual Cues

In *Nomadic Radio*, context dynamically changes the system's operational modes and scales the notifications for incoming messages. Contextual cues are determined by observing several aspects of the information delivered, state of the user and the environment. The primary contextual cues used include: *message priority level* from email filtering, *usage level* based on time since last user action, and the *conversation level* estimated from real-time analysis of environmental audio. We will now consider the mechanisms for sensing such contextual cues as described below:

5.6.1 Priority Level

The priority of incoming messages is explicitly determined via content-based email filtering using CLUES [Marx 95, 96b], a filtering and prioritization system. CLUES has been integrated in *Nomadic Radio* to determine the timely nature of messages by finding correlation between a user's calendar, rolodex, to-do list, as well as a record of outgoing messages and phone calls. When CLUES finds an area code in a calendar entry, it checks the rolodex for people who share that area code. Email messages arriving from people are prioritized if the user is traveling and meeting others in the same geographic area (based on area code entries in their calendar). This provides an implicit form of location awareness (a real-time approach is discussed in the future work section).

A remote server process runs CLUES on an hourly basis and automatically generates filtering rules from the user's personal data. These rules are integrated with static rules created by the user for prioritizing specific people or message subjects (defined in the user's *.procmailrc* file on the server). When a new email message arrives, keywords from its sender and subject header information is correlated with static and generated filtering rules to assign a priority level to the message. The current priorities include: group, personal, very important, most important, and timely. The remote augments message header information with priority level when notifying *Nomadic Radio*. By default, voice messages receive most important priority whereas news broadcasts receive group priority. If the user maintains a file with preferred caller numbers and preferred times of day when she likes to hear news summaries, such information generates appropriate priorities for notification. Alternatively, the system should track when a user most frequently activates news summaries or listens to voice messages from certain callers. This would allow an implicit means for learning a notification strategy for prioritizing voice messages and broadcasting news summaries at appropriate times of day.

5.6.2 Usage Level

A user's last interaction with the device determines their usage level. If users are engaged in voice commands to the system or browsing messages on it (or have been in the last few minutes), they are probably more inclined to hear new notifications and speech feedback. Every action of the user, while using *Nomadic Radio,* is time-stamped. A timer continuously compares the current time to that of the last user action to dynamically change modes (described in the next section). When messages arrive the last action time is used to determine the current usage level; this is used in the notification model discussed later. One problem with using last actions for setting usage levels is that if a user deactivates an annoying message, that action is again time-stamped. Such negative reinforcements continue to increase the usage level and the related notification. Therefore negative actions such as stopping audio playback or deactivating speech, are excluded from generating actions for computing the usage level.

5.6.3 Conversation Level

Conversation in the environment can be used to gauge whether the user is in a social context where an interruption is less appropriate. If the system detects the occurrence of more than several speakers over a period of time (10-30 seconds), that is a clear indication of a conversational situation. Two different approaches were developed for extracting contextual cues from environmental sounds.

An earlier approach based on prior work at the Machine Listening Group at the MIT Media Lab [Arnaud95] was designed to extract and classify features from 5 pre-defined classes of sounds in the environment [Sawhney97]. The focus of the work was on distinguishing longer-duration environmental sounds into pre-defined classes using near real-time classification techniques. Environmental sounds were recorded via DAT and segmented into training and test data-sets. The auditory front-end extraction utilized a Gammatone filter bank defined by Patterson and Holdsworth for simulating the cochlea. This was previously implemented within the *Auditory Toolbox* in Matlab. We chose to use 21 frequency bands extending from a range of 80 Hz to 8000 Hz. The Hilbert transform extracts an energy envelope of the signal. A nearest neighbor estimator using a Euclidean distance metric classified test data relative to that of the 5 training classes. Preliminary results showed an overall accuracy in classification at 68% (voice was classified at 80%), yet the system was not implemented to run in real-time. Hence, a 15 second audio sample requiring nearly 60 seconds for extraction and analysis is considered too slow for use in a adaptive notification model or practical integration on a wearable system.

Recent collaboration with Brian Clarkson at the vision and modeling group (Vismod) at the Media Lab, allowed integration of a real-time audio classifier in *Nomadic Radio* that extracts the level of speech from environmental sound [Clarkson98]. The system is based on Hidden Markov Models (HMM's) [Rabiner89], a statistical/pattern recognition framework. Audio data sampled at 16 kHz and quantized to 16 bits, is analyzed with a 256 point FFT giving a vector of frequencies extracted at each time step. To make the system robust for widely varying noise levels, frequencies are normalized so that later processing is sensitive only to relative frequency levels. This effectively allows us to use the same models for loud speech and soft speech.

In the second stage, there is a collection of *fully connected HMMs* to model the behavior of the spectrogram over short periods of time (approx. 0.1-1.0 seconds). The number of states used, varies between 1 and 15 depending on the complexity and duration of the sound being modeled. HMMs are trained to detect the presence of the formants (or pitch tracks) in voiced speech. These HMMs learn how rapidly the formants change (rate of speech) and what frequencies they cover (speaker's pitch). During classification, these HMMs detect the recurrence of similar formant behavior. Specifically, an HMM yields the probability that the model could have matched the input sequence (see figure 5.5).

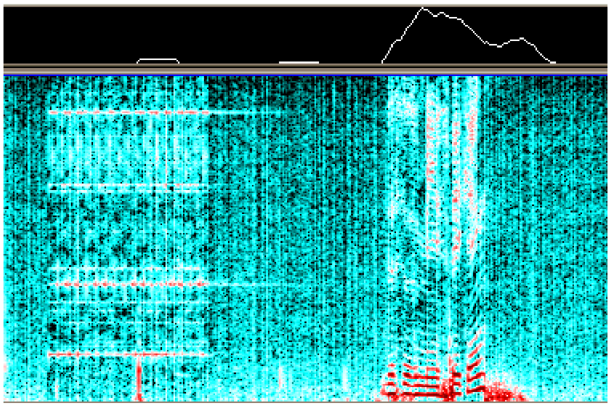


Figure 5.5: Bottom panel shows a spectrogram (~ 4 secs) containing telephone ringing and speech utterance. The top panel is the output probability (log likelihood) of an HMM trained on speech. The HMM correctly identified the speech (courtesy of Brian Clarkson)

The pitches (and how they move through time) are dependent on the speaker. Therefore, it is possible to include some speakers and exclude others in the measurement of the speech level in the environment. Yet, it is necessary to have separate models for a number of different female and male speakers to detect any speech regardless of the speaker. The current implementation of the audio classifier runs in real-time on a Pentium PC. We look at a window of time (5-10 seconds) during which a message comes in to determine the current conversational level. Integration of the audio classifier with *Nomadic Radio* will be further discussed in chapter 7. At the time of writing, refinement and evaluation of the classifier's performance, is being conducted. In section 5.8 we describe how these contextual cues are parameterized for computing scale and latency in notifications. In future work (chapter 7), we consider how contextual cues are inferred by classifying other non-speech sounds in the environment and by using techniques for location awareness.

5.7 Dynamic Operational Modes

Nomadic Radio runs under several dynamically changing modes of operation. The system continuously degrades its auditory interface services based on time elapsed since the last user action. This transition is accelerated if a high level of conversation is observed for a few minutes. A dynamic state model (see figure 5.6) allows the system to transition to states with minimum auditory interruption. Such states consequently require lower memory usage and optimize performance. The system generally requires 25-40% processing resources depending on the number of text and audio messages (instantiated as objects) in memory and the number of concurrently active program threads. On a wearable system, such resources are limited and must be reduced when possible i.e. when the user is inferred to be inattentive. The system is initially launched in *active speech mode*, with no visual display. The visual display is turned on or off explicitly by the user and is automatically deactivated after 2 minutes of inactivity. In *active speech mode*, speech feedback is scaled down over time from full message playback to*preview* (1/5th of message delivered in a time-compressed form) and finally to *summary* (only message header spoken). After a few minutes of inactivity, the system switches to *auditory mode* (non-speech) by turning off speech synthesis and recognition. In such states, the system continues monitoring audio for a "wake up" utterance from the user, and relies on auditory cues to convey feedback and notifications.

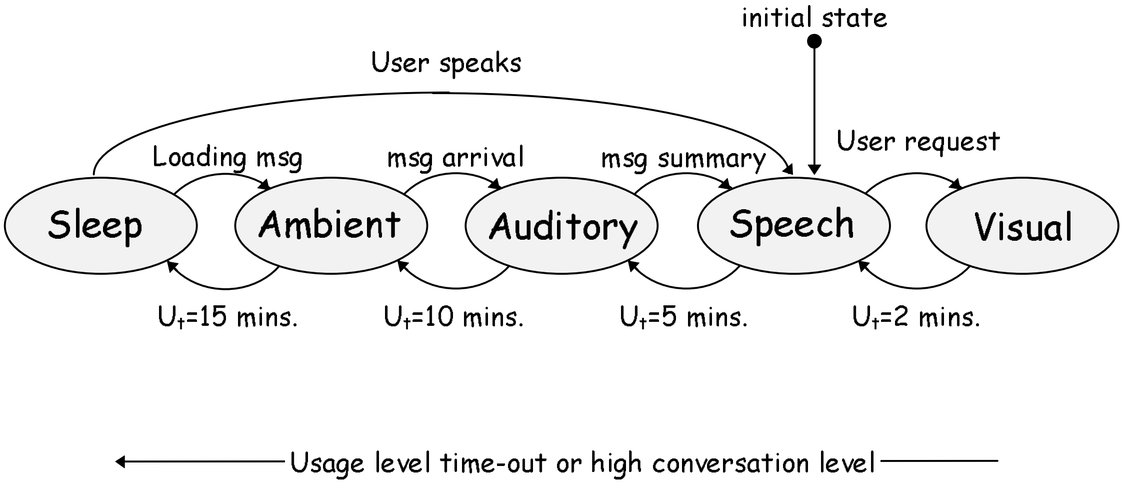


Figure 5.6: State-transition diagram with transitions activated by contextual cues. Here U_t represents the default deactivation time (in minutes) since last user action

Eventually the system switches to an *ambient mode* where most auditory cues are turned off, but awareness is provided via continuous ambient sounds. After 15 minutes of inactivity from the user, the system enters *sleep mode* (1% CPU usage) which conserves power and provides the least interruption if the user has not been paying attention. At any state, the system switches to higher (or lower) modes when a new message arrives based on its inferred priority level or if a valid speech command is detected.

The default settings for usage level time-outs on each transition are modified via parameter settings on a visual interface. During state transitions, the user is notified of the transition via a combination of auditory cues and synthetic speech feedback. For example, switching out of visual and speech modes is indicated by distinct auditory cues. An ambient state to silence transition is implicitly indicated by a noticeable lack of any background ambient sound. If the user has previously selected high level of speech feedback, the system will also announce transitions as "deactivating visual display" or "Nitin, you are not paying attention, so I won't speak anymore" (while switching to auditory cues mode). Transitions to ambient and sleep modes are not accompanied with any spoken feedback, since that would require switching back to the speech mode first and would pose an undesirable interruption to the listener.

5.8 Contextual Notification Model

Deciding when and how to interrupt the listener allows a system to provide effective and unobtrusive notification. Contextual notification is based on three factors: *message priority level*, *usage level*, and *conversation level* estimated from real-time analysis of environmental audio. These parameters are weighted for high, medium or low interruption levels set by the user. As messages arrive, an appropriate notification level is selected based on a computed average of these weighted parameters [Papp96]. The model also computes a notification *latency* i.e. a period of time to wait before playing the message, after a notification is delivered. The message notifications are shown on a visual graph where they can be interactively manipulated by the user. The user dynamically reassigns new weights to the usage, conversation or priority parameters such that the model provides an appropriate and predictable notification response for specific situations. In addition, the user's actions during notification provide positive and negative reinforcement of the model scaling the notifications dynamically over time.

5.8.1 Computing Notification Scale and Latency

The notification model must rely on three contextual cues: priority, usage and speech level to decide when and how to present an incoming message. A numerical cost-value approach is used in this model to represent the (1) value of information to be delivered and (2) the cost of interruption based on user and environmental parameters. Hence, the input dimensions for this model must be converted from relative discrete levels to continuous values, and parameterized as follows:

Priority Level: Priorities are obtained by filtering email via CLUES and the predefined preferences for the user. There are 5 discrete priority levels that apply to email: *group*, *personal*, *timely*, *very important*, and *most important*. Voice messages are assigned "most important" if the caller number is known, else it is assigned "very important". News broadcasts are assigned a "group" priority by default. Yet, if the news message is broadcast at a user's preferred listening time (defined his preference file), the message is assigned a *personal* priority. Calendar events are assigned to one of two priorities: *weekly event* or *special event*. All messages are assigned a priority i = 2 to 8, with uncategorized messages assigned to priority i

=1. These priorities are parameterized by logarithmically scaling each of the eight priorities ($\text{Priority Levels}_{\text{Max}} = 8$) within a range of 0 to 1, giving a $\text{Priority}_{\text{Level}}\ (\ i\ )$. Logarithmic scaling ensures that higher priority messages are weighted higher relative to the low priority messages.

$$\text{Priority}_{\text{Level}}\ (\ i\ ) = (\ \log\ (\ i\ ) / \log\ (\text{Priority Levels}_{\text{Max}}\ )\ )$$

$$\text{where Priority } i = \{\ 1\ ..\ \text{Priority Levels}_{\text{Max}} = 8\}$$

Usage Level: Usage is measured based on a scaled value of the time since the last user action. Each user action is time-stamped and an active timer compares the time since the last action with default values for state transitions at every clock tick. When a new message arrives, its time of arrival is compared with the $\text{Last Action}_{\text{Time}}$ and scaled based on the $\text{Sleep}_{\text{Time}}$ (default at 15 minutes). High usage is indicated by values closer to 1 and any message arriving after $\text{Sleep}_{\text{Time}}$ are assigned a zero usage level. Logarithmic scaling ensures that there is less variance in usage values for recent actions relative to usage levels computed for any duration (of zero activity) closer to the $\text{Sleep}_{\text{Time}}$. The user not responding for 10-60 seconds, has less effect on notification than the user not responding for over 10-15 minutes.

$$\text{Idle}_{\text{Time}} = \log\ (\ \text{Current}_{\text{Time}} - \ \text{Last Action}_{\text{Time}}\ )$$

$$\text{Usage}_{\text{Level}} = (\ (\log\ (\text{Sleep}_{\text{Time}}) - \text{Idle}_{\text{Time}}\ ) / \log\ (\text{Sleep}_{\text{Time}})\ )$$

Conversation Level: When a message arrives, the system polls the audio classifier for speech level on a pre-defined window of time (default at 5 seconds). The likelihood of speech detected in the environment (a probability) is computed every millisecond (1 frame) and averaged for each frame over the entire window specified ($\text{Window}_{\text{size}}$ in secs.). In addition, the probabilities are weighted, such that most recent time periods in the audio window are considered more relevant in computing the overall $\text{Speech}_{\text{Level}}$.

$$\text{Speech}_{\text{Level}} = (\ \Sigma_{\text{frames}}\ p\ (\ \text{Speech}_{\text{Class}}\ )\ \ x\ \text{frame}_{\text{wt}}\ ) / \text{frames}\ )$$

$$\text{where frames} = \{\ 1\ ..\ \text{Window}_{\text{size}}\ x\ 1000\ \}$$

Notification Level: Values for each contextual cue are weighted appropriately to change their relative influence on the overall notification level. Thus, a weighted average for all three contextual cues provides an overall $\text{Notification}_{\text{Level}}$. The conversation level ($\text{Speech}_{\text{Level}}$) has an inversely proportional relationship with notification i.e. a lower notification must be provided during high conversation. This aspect is taken into account in the weighted average.

$$\text{Notification}_{\text{Level}} = ((\text{Priority}_{\text{Level}}\ x\ P_{\text{wt}}) + (\text{Usage}_{\text{Level}}\ x\ U_{\text{wt}}) + (\ (1 - \text{Speech}_{\text{Level}})\ x\ S_{\text{wt}})) / 3$$

$$\text{where } P_{\text{wt}},\ U_{\text{wt}} \text{ and } S_{\text{wt}} \text{ are weights for Priority, Usage and Conversation levels}$$

This $\text{Notification}_{\text{Level}}$ must be translated to a discrete notification scale to play the message. There are currently 7 notification scales described further in section 5.5 (ordered in descending order): *foreground, full message, preview, summary, audio cue, ambient, silence* ($\text{Notify Levels}_{\text{Max}} = 7$). The $\text{Notification}_{\text{Level}}$ computed must be compared to the thresholds for each of 7 scales to play the message appropriately.

The $\text{Notify Levels}_{\text{Max}}$ are scaled by two to produce thresholds with a greater range to accommodate the notification levels computed under varying interruption levels. This provides a reasonable $\text{Notification}_{\text{Scale}}$ for each message.

$$\text{Threshold}_{\text{Level}} (i) = (\log (i) \ / \ \log (\text{Notify Levels}_{\text{Max}} \times 2))$$

$$\text{If} \ (\text{Notification}_{\text{Level}} (i) > \text{Threshold}_{\text{Level}} (i)) \text{ then assign } \text{Notification}_{\text{Scale}} = i$$

$$\text{where } i = \{ 1 .. \text{Notify Levels}_{\text{Max}} = 7 \}$$

Latency: This represents the period of time to wait before playing the message to the listener, after a notification cue is delivered. Latency is computed as a function of the notification level and the maximum window of time ($\text{Latency}_{\text{Max}}$) that a lowest priority message can be delayed for playback. The default Max is set to 20 seconds, but can be modified by the user.

$$\text{Latency}_{\text{Level}} (i) = (1 - \text{Notification}_{\text{Level}} (i)) \times \text{Latency}_{\text{Max}}$$

A higher $\text{Notification}_{\text{Level}}$ will cause a shorter latency in message playback and vice versa. An important message will play as a "preview" within 3-4 seconds of arrival, whereas a group message may play as a "summary" after 11-13 seconds of arrival (given high usage and low conversation levels).

5.8.2 Reinforcement Strategies for Notification

The overall notification levels for the model presented are based on pre-defined weights for priority, usage and conversation levels. These weights are initially set to high, medium, or low in the start-up settings file. These three weight settings have been selected by experimenting with the notifications in the system. The system provides the user with three different mechanisms to adjust these weights.

Pre-defined Interruption Levels

The default weights are changed by the user via a spoken utterance or button command "Set {High | Medium | Low} Interruption". For each of the three levels, the default weights are simply defined by experimenting with several settings for the weights. They provide an approximate behavior for notifications and help bootstrap the system for novice users. The last weight-set assigned by the user is saved on the server when the system exits, and provides her a *portable interruption profile* for future use.

Interactive Visual Reinforcement

It is clear that the user would wish to make finer adjustments to these weights, tuning the notification levels that seem satisfactory to her messaging environment and personal listening preferences. However, hand-tuning such weights requires changes to a complex set of inter-related parameters. In addition, users will not feel comfortable to *trust* delegating message filtering and notification to a system that does not show its notification strategy clearly [Maes94]. Hence, *Nomadic Radio* provides a visualization of the notification levels based on the current weights and allows users to change these weights dynamically. As messages arrive the system plots the notification level on graphs depicting priority vs. usage level and priority vs. conversation level. The graph is similar to dynamic Starfield display used in several interactive visual querying applications [Ahlberg94].

When no messages are shown, the graphs display predicted notification levels as slopes based on the two graph axis. See figure 5.7 for two example displays showing the notification graph with "Medium" and "High" weights assigned. The messages on the top are higher priority ones (*very important*), whereas the ones shown on the lowest slope indicate lower priority *group* messages. The computed notification level places the messages above specific notification thresholds and scales their playback accordingly. Over time, the notifications for all messages are shown to degrade as the usage level goes down or conversation level increases.

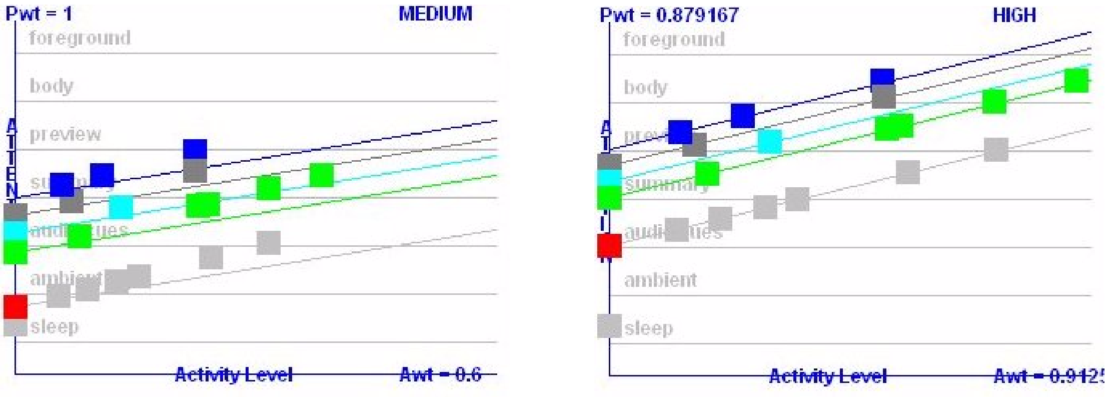


Figure 5.7: Dynamic notification graph showing notification assigned to messages over time based on priority and user activity level.

These default weights can be further adjusted by the user for different priority (P_{wt} = 0.879) and usage (U_{wt} or A_{wt} = 0.942) weights as shown in the "High" weight display. The user interface provides a simple means for modifying and visualizing a large number of weights dynamically. As the user drags her pointer on the display, across any of the axis ($P_{wt,}$ A_{wt} or S_{wt}), the slopes move accordingly. In addition, new notification levels are re-computed for all messages in real-time to provide a dynamic visualization of the change in weights. This provides an interactive form of reinforcement for the notification model, while allowing the user to easily visualize the consequence of different weighting strategies. Using such a technique, users may be able get a better understanding of how their new weighting strategies will influence future interruption based on observing the changing effect on prior messages. This approach may not be appropriate for novice users, who may prefer to use pre-defined weights until they feel the need to make finer adjustments later.

Adaptive Reinforcement of Notifications

The above approaches for setting default interruption levels and interactively change weight assignments, requires explicit and intentional reinforcement. A simpler approach allows users to change interruption and notification levels by their implicit actions while playing or ignoring messages.

The system allows users positive and negative reinforcement by monitoring their actions during notifications. As a message arrives, the system plays an auditory cue if its computed notification level is above the necessary threshold for auditory cues. It then uses the computed latency interval to wait before playing the appropriate summary or preview of the message. During that time, the user can request the message played earlier or abort any further notification for the message via speech or button commands. If aborted, all weights are reduced by a fixed percentage (default is 5%). This is considered a negative reinforcement. If the user activates the message before the notification, the message playback scale selected by the user, is used to increase all weights. If the message is ignored, no change is made to the weights, but the message remains active for 60 seconds during which the user's actions can continue to influence the weights. Hence, 15 seconds after hearing a notification, a user may play a summary, and the weights would be upgraded accordingly. Let's revisit the scenario we introduced at the beginning of this section to examine how reinforcement is used to change notification levels over time.

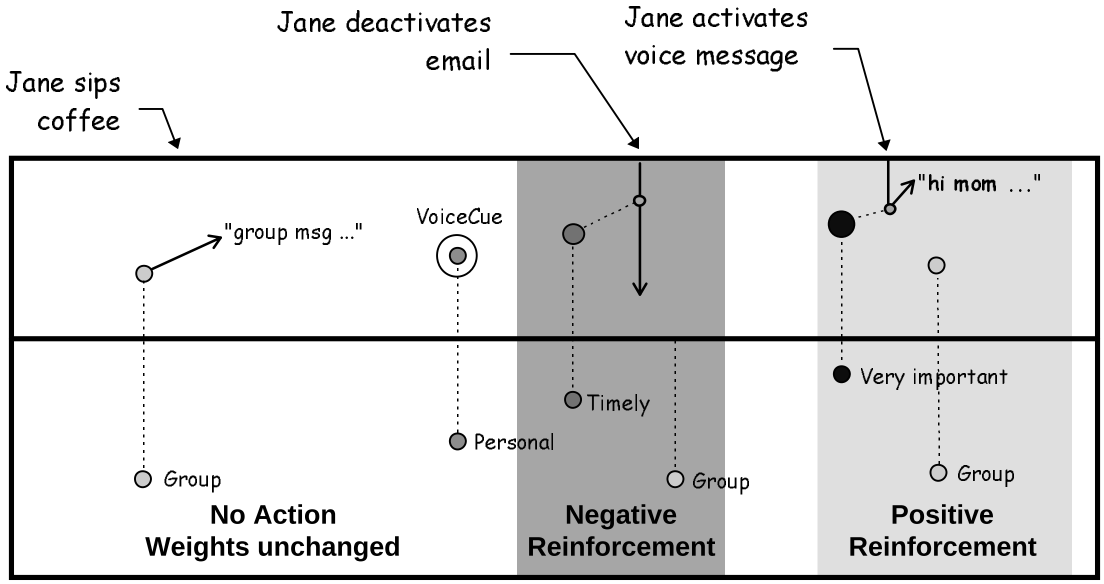
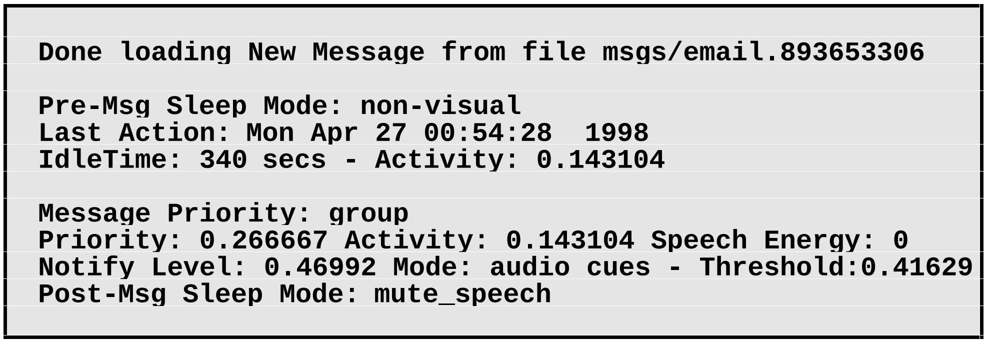


Figure 5.8: Reinforcement strategies used by Jane while listening to her messages.

Figure 5.8 shows a zoomed in view of the extended scenario in figure 5.2, focusing on Jane's actions for that reinforced the model. Jane received several messages during her meeting. She ignored most of the group messages as well as the personal message from Susan (the weights remain unchanged). Before her meeting, she did listen to a summary from a group message. Since she took no action after hearing this message within 60 seconds, its priority level was set to one originally computed (as summary). For the timely message, Jane interrupted the message before it played to abort the playback. This reduced the weights for future messages, and the ones with low priority (group message) were not notified to Jane. The voice message from Kathy, her daughter, prompted Jane to reinforce the message by playing it. In this case, the weights were increased. Jane was notified of a group message shortly after the voice message, since the system detected higher usage activity. The system correctly scaled down notifications when Jane did not want to be bothered (negative reinforcement) and notifications were scaled up when Jane started to use the system to browse her messages again (positive reinforcement).

Now lets consider two actual examples of notification levels and latency computed for email messages with different priorities.

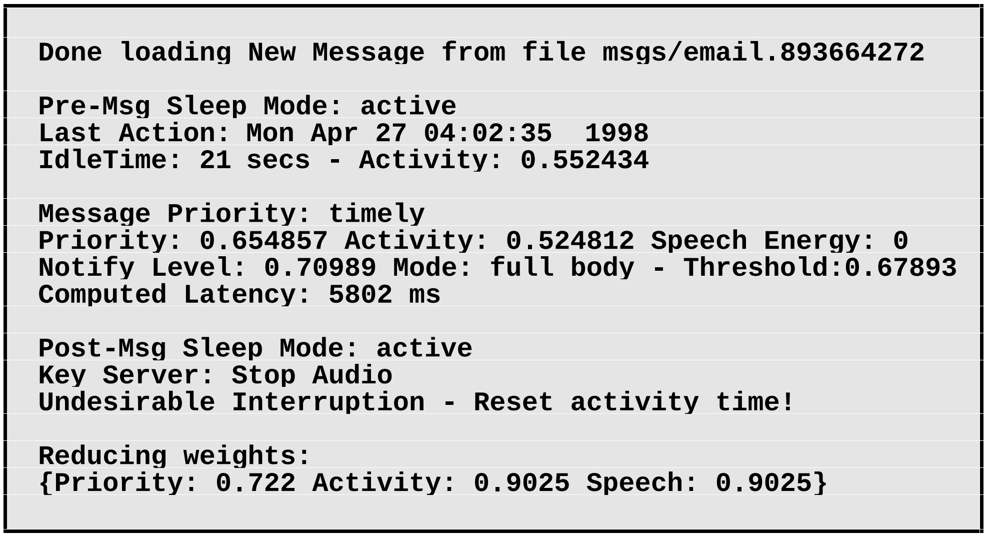


```
Done loading New Message from file msgs/email.893653306

Pre-Msg Sleep Mode: non-visual
Last Action: Mon Apr 27 00:54:28  1998
IdleTime: 340 secs - Activity: 0.143104

Message Priority: group
Priority: 0.266667 Activity: 0.143104 Speech Energy: 0
Notify Level: 0.46992 Mode: audio cues - Threshold:0.41629
Post-Msg Sleep Mode: mute_speech
```

Figure 5.9: Notification level computed for a group email message. The user has been idle for some time and the message is heard as an auditory cue. The system also switches from non-visual to mute_speech (or auditory) operational state.



```
Done loading New Message from file msgs/email.893664272

Pre-Msg Sleep Mode: active
Last Action: Mon Apr 27 04:02:35  1998
IdleTime: 21 secs - Activity: 0.552434

Message Priority: timely
Priority: 0.654857 Activity: 0.524812 Speech Energy: 0
Notify Level: 0.70989 Mode: full body - Threshold:0.67893
Computed Latency: 5802 ms

Post-Msg Sleep Mode: active
Key Server: Stop Audio
Undesirable Interruption - Reset activity time!

Reducing weights:
{Priority: 0.722 Activity: 0.9025 Speech: 0.9025}
```

Figure 5.10: Notification level and latency computed for a timely email message. The user has used the system recently and thus the message is scheduled to play as a summary in approx. 6 secs. The user’s action of stopping audio before it plays induces a negative reinforcement, reducing all the current weights.

Figure 5.10 shows an example of an actual email message that arrived while the system was in *active speech* mode. The activity level was high and conversation level low. As a timely message, it received a priority and consequently a notification level higher than the threshold for summary playback. A moderate latency time was chosen, but it is clear that the user interrupted the notification by a button press, thereby aborting the summary playback. The user's action also reduced the overall weights by 5%. The weights were previously assigned to "High" interruption levels.

Continuous reinforcement over time should allow the system to reach a state where it is somewhat stable and robust in predicting the user's preferred notification level and latency. Any evaluation of such a model would measure the errors in prediction as observed by comparing the computed notification level relative to that set by the user's actions. In this scenario, the system is optimized over time by trying to reduce the least square errors in weights for a set of reinforcements by the user (say a sliding window of 50-100). Over time such optimization may show some convergence. Yet a user's listening patterns may change on a daily or weekly basis and this would require collection of data over several weeks to provide any reasonable adaptive optimization in notification strategy.

5.8.3 Evaluating the Notification Model

To evaluate the dynamic operation of the contextual model, we must observe the effect of contextual cues on the scale of notification selected, over time. In *Nomadic Radio*, for each incoming message an *action* is recorded along with the level of contextual cues, the user's response and the level of audio being played during message arrival. The action information can be saved on the server and evaluated on an off-line basis in a visualization package such as Matlab. An extensive evaluation requires collection of notification data over several weeks. We will consider a preliminary evaluation based on sample data collected over a period of a few days.

Capturing User Actions

When an incoming message is delivered, the levels of three contextual cues are observed i.e. priority, activity and speech. As described earlier, a weighted average of these levels is used to compute the notification. If the user responds to the message by stopping the audio or upgrading the notification, that action is scaled (from 0-1) and recorded along with the response time (in minutes). In addition, the *audio bandwidth* is computed based on all audio streams currently playing on *Nomadic Radio* while the message is delivered (abD) and/or played (abP). These parameters are recorded (see figure 5.11) for possible correlation during off-line analysis of the data. Finally, the delay time is the time since first action; this primarily is used to visualize all the messages on a scale based on their time of arrival.

```
Last Action 26:
[Thu Apr 30 21:08:01  1998]  { Delay: 20.8036 Priority: 0.333333 Activity: 0 Speech: 0
Notify: 0.264153 Action: 0 rTime: 0 abD: 0 abP: 209.997}
```

```
26 ACTIONS CURRENTLY STORED

{ Date | Delay | Priority | Activity | Speech | Notify | Action |  rTime | abD | abP }

[Apr 28 01:29:09]  { 1.15    1       0       0     0.57      0        0      0      0}
[Apr 28 04:34:33]  { 4.24    0.33    0       0     0.40      0        0      0      0 }
[Apr 28 06:08:26]  { 5.81    0.33    0       0     0.40      0        0      0      209.97}
[Apr 28 10:49:01]  { 10.48   0.33    0.59    0     0.58      0        0      16.42  0 }
[Apr 28 11:07:58]  { 10.80   0.33    0.06    0     0.29      0        0      0      41.99}
[Apr 28 11:23:35]  { 11.06   0.33    0       0     0.27      0.5      9.78   0      0}
[Apr 28 11:28:19]  { 11.14   1       0.21    0     0.55      0        0      0      0}
[Apr 28 11:54:47]  { 11.58   1       0       0     0.50      0.83     12.7   0      0}
[Apr 28 11:54:51]  { 11.58   1       1       0     0.73      0.83     0.05   0      0}
[Apr 28 11:55:55]  { 11.60   1       1       0     0.75      0        0      0      0}
[Apr 28 12:00:03]  { 11.67   1       0.20    0     0.57      0        0      0      0}
[Apr 28 12:24:07]  { 12.07   1       0.21    0     0.57      0        0      0      0}
[Apr 28 15:21:20]  { 15.02   1       0       0     0.52      0        0      2.5    0}
[Apr 28 15:48:40]  { 15.48   0.33    0.25    0     0.36      0        0      0      41.99}
[Apr 28 15:55:03]  { 15.58   0.86    0.42    0     0.53      0        0.18   0      0}
[Apr 28 16:04:05]  { 15.73   0.33    0       0     0.29      0        0      0      0}
[Apr 29 13:21:27]  { 13.02   0.77    0       0     0.53      0        0      0      0}
[Apr 29 13:39:16]  { 13.32   0.86    0       0     0.56      0        0      0      0}
[Apr 29 14:21:14]  { 14.02   0.33    0       0     0.42      0        0      0      2.5}
[Apr 29 14:40:30]  { 14.34   0.86    0.44    0     0.66      0        0.45   0      0}
[Apr 29 15:27:47]  { 15.13   0.33    0       0     0.40      0        0      4      4}
[Apr 29 15:36:41]  { 15.28   0.86    0.43    0     0.65      0        0.3    0      0}
[Apr 29 15:56:10]  { 15.60   0.86    0.43    0     0.54      0        0.4    0      0}
[Apr 30 16:00:58]  { 15.68   1       0       0     0.43      0.83     0.37   3      0}
[Apr 30 20:08:22]  { 19.80   0.33    0.27    0     0.34      0        0      419.99 209.99}
[Apr 30 21:08:01]  { 20.80   0.33    0       0     0.26      0        0      0      209.99}
```

```
Saving Actions on host: ml - port: 2007
Connected to 18.85.13.107/18.85.13.107:2007
26 Actions Saved!
```

Figure 5.11: Actions are recorded for each message based on user response or lack thereof within 60 seconds. These actions are time-stamped and periodically saved on the server for off-line analysis. Here 26 actual actions are captured and later plotted.

The data can be glanced to see a few trends: messages with higher priorities are assigned a high notification level, e.g. message 1 has a priority of 1 unit and a notify level of 0.57. However, messages with both high priority and high activity are presented with much higher notification, e.g. message on [Apr 29 14:40:30] shows priority of 0.86 and activity level of 0.44, hence its notification level is 0.66.

The highlighted area in the data set shows an example of two messages arriving within seconds of each other. In the first message, the user responds to the message within 12.7 seconds, increasing the activity level from 0 to 1 for the next message. Hence, when the next message arrives its notification level is computed to be higher (0.73). This data can also be visualized to provide a high level view of the model.

Data Visualization

The following plots were generated in *Matlab*. These plots compare only three attributes: Activity Level, Priority Level and the subsequent Notification, over time. Plotting these attributes of the messages over time, provides an easy way to verify the behavior of the notification model. It is expected that the model will increase notifications for messages with higher priority when the user has clearly been interacting with the system recently (indicated by high usage level). The model should rapidly decay notifications for messages arriving after a few minutes of inactivity (due to logarithmic scaling). However, even during extended periods of inactivity, the model should be responsive to allow high priority messages to be filtered through to the user, by providing a reasonable level of notification for those messages. Lets consider some examples in the graphs shown below.

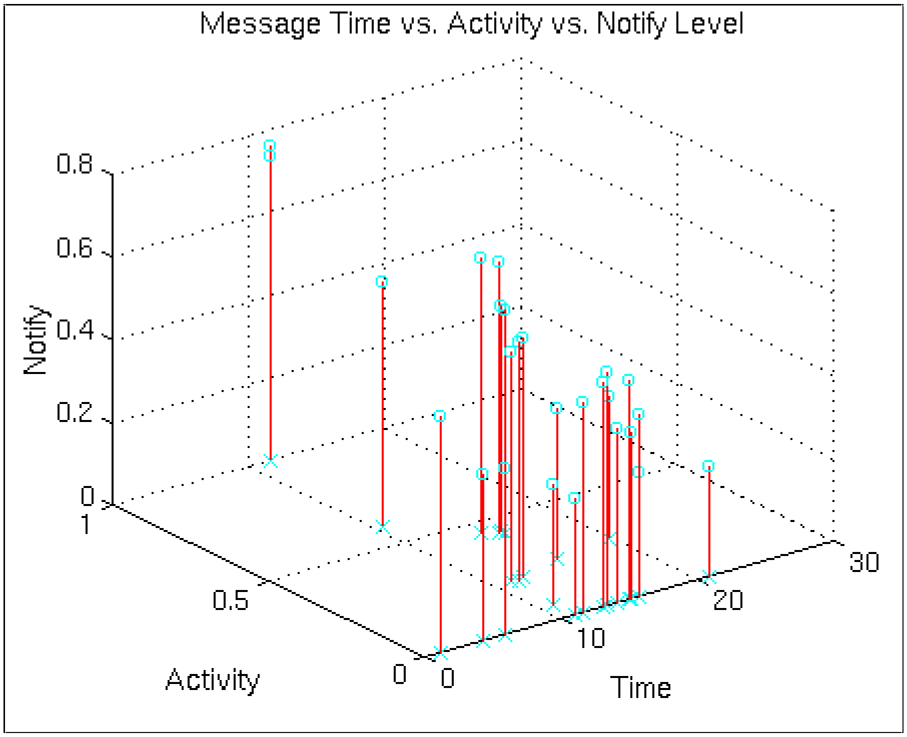


Figure 5.12: A graph showing change in the notification based on activity level of the user over time. High notification is selected for higher activity levels.

In figure 5.12, the activity level of messages arriving over time show a few basic trends. The messages arriving after the system goes to sleep mode (past 15 minutes, shown by messages at Activity = 0) were all assigned lower notification levels relative to those arriving at moderate activity levels (Activity ~ 0.5 or upto 7 minutes after the user last responded). Clearly, two messages that arrived while the user was actively using the system, were assigned the highest notifications. One potential outlier observed in this graph for a message arriving at the beginning (Time ~ 0 and Activity = 0), seems inconsistent with the predicted behavior of the model. Yet, this can be explained by observing (message 2 in the data set) that the priority level for that message was set to 1 (the highest level) influencing its high notification level computed.

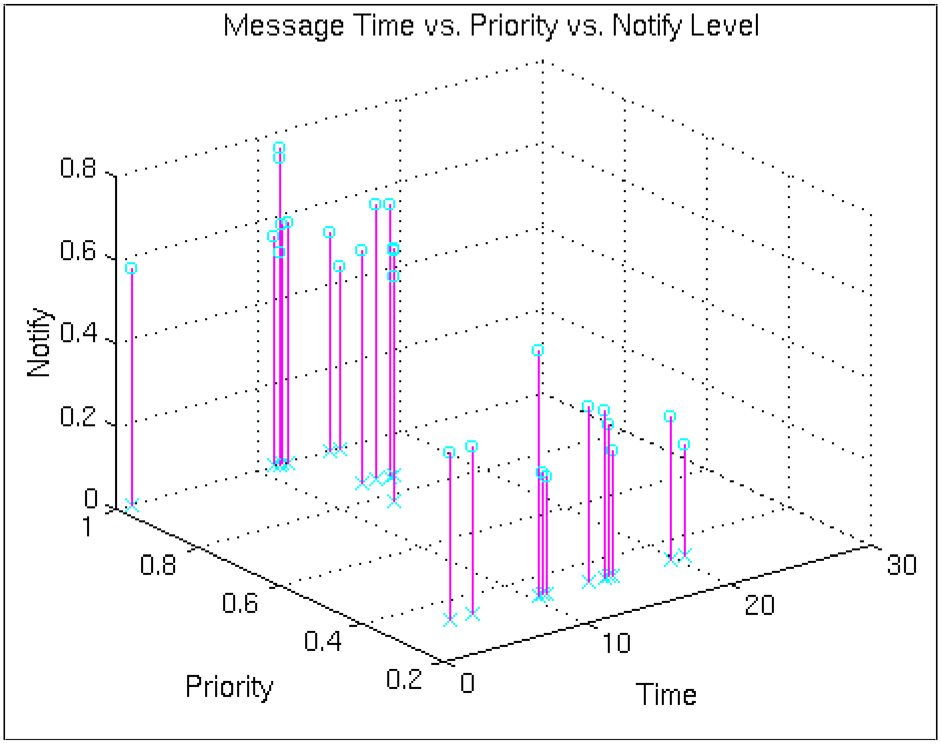


Figure 5.13: A graph showing change in the notification based on activity level of the user over time. High notification is selected for higher priority levels.

Figure 5.13 depicts priority vs. notification for messages arriving over time and shows a more consistent behavior. All messages with high priority (between 0.8-1 units) have relatively higher notifications, whereas ones below 0.4 units are clearly computed with lower notification levels. Here again one potential outlier (around ~10 minutes in time) seems higher than most at its level. If compared with the graph in figure 1, it can be associated with the message having a high activity level and hence higher notification. Thus a complete picture can only be obtained by directly comparing notifications of messages based on both activity and priority, as shown in figure 5.14.

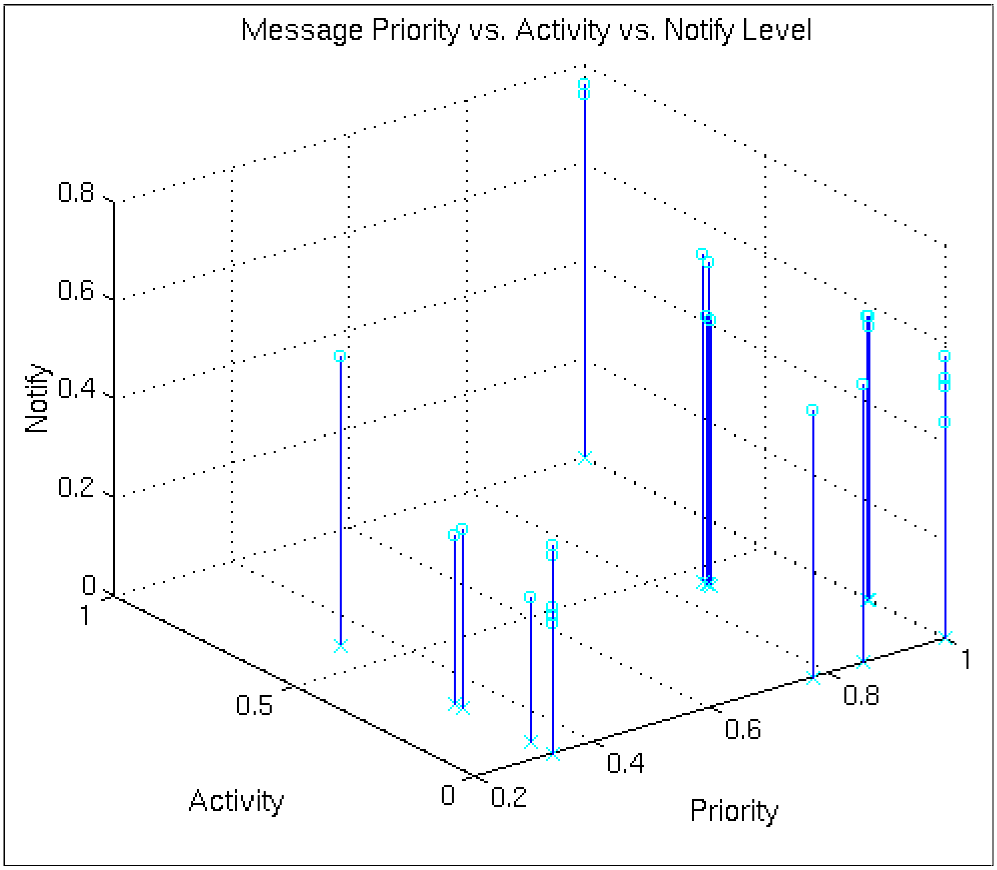


Figure 5.14: This graph shows a combined view of notification based on priority and activity levels. High notification is selected for higher priority levels (past 0.8 units). Low notification can be seen corresponding with low activity and priority levels (~0.5 units).

In figure 5.14, high priority messages are set to high notification levels even though the activity was lower for those messages. One high activity message (over 0.5 units) was assigned a higher notification even though its priority was low, which is consistent with the model. Finally, messages with highest priority (Priority = 1) and activity levels (Activity = 1) were assigned the highest notification (2 messages at the inner most corner of the graph).

This preliminary evaluation represents only a small dataset (26 data points over 3 days), yet it demonstrates reliable and predictable computed outcomes from the contextual notification model. When conversational cues are fully integrated, we can evaluate an additional dimension of the model which will surely influence its usage in a social environment. The model has been demonstrated for scaling complex notifications in *Nomadic Radio*, but it is foreseeable that such a model can be easily applied to everyday personal devices such as cell-phones, pagers and PDAs.

6. System Architecture

In the *Nomadic Radio* architecture, interface components such as spatialized audio, speech synthesis, recognition, tactile input and audio monitoring provide local and distributed services to wearable and wireless platforms. The architecture is modular and extensible such that users can create and subscribe to new services or continue using the system reliably even when some services are unavailable. We will discuss techniques for presenting audio streams synchronized with other services using a multi-threaded design. The user's personal messages and application data and must be handled in a reliable and flexible manner within such a distributed architecture. This chapter discusses the approach used for robust access, coordination and distribution of these information sources and audio interface services for effective use in a nomadic environment.

6.1 Unified Client-Server Messaging Architecture

Timely messaging and remote information access requires a nomadic computing infrastructure. In the Speech Interface group at the MIT Media Lab, we have developed an environment [Schmandt93] which allows subscribers to access information such as email, voice messages, weather, hourly news and calendar events using a variety of interfaces such as desktops, telephones, pagers, fax, etc. One of the objectives for*Nomadic Radio* is to provide direct access to the information services in this infrastructure on a wearable audio platform. For wearable access, such services are unified in a manner that is scaleable as new information sources are added (or subscribed to by the user) and easy to navigate using an audio-only modality.

Nomadic Radio consists of client and remote server components that communicate over the wireless LAN. The current architecture (shown in figure 6.1) relies on server processes (written in C and Perl), running on Sun Sparcstations, which utilize the telephony infrastructure in the Media Lab's Speech Interface group. The servers are designed to extract information and provide remote access to services including voice mail, email, hourly updates of ABC News, and personal calendar events. The *Nomadic Clients* when notified by the *Radio Server* (both written in Java to operate on PC-based wearable platforms), download the appropriate text or audio files stored on the web server. The different servers and remote processes are described below:

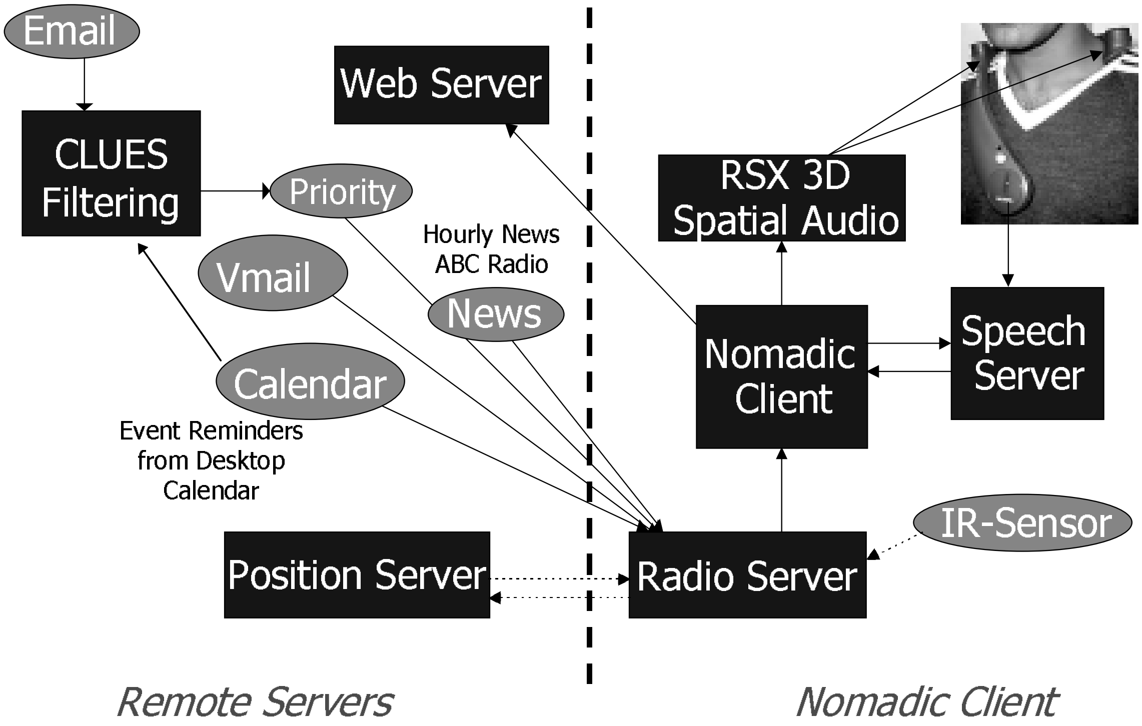


Figure 6.1: The architecture of *Nomadic Radio*, showing communication between remote server processes and the *Nomadic Client.* This architecture shows a "wired" approach, with speech services operating on the wearable itself. A "wireless" approach allows speech recognition and audio monitoring distributed on remote machines. The *Position Server* and IR-based sensors for location awareness are currently being integrated.

6.1.1 Distributed Access via a Web Server

An *Apache* web server, running on a SUN Sparcstation in the Speech Interface Group, provides remote access to user messages, information services and application-specific data. A web-based approach allows ubiquitous views of messages for*Nomadic Clients*, either running as Java applets on web-browsers or as independent applications on Wearable PCs. The data on the web server is structured in the following manner. Each user of *Nomadic Radio* has an individual directory where all her personal messages and preference files are stored. The web server allows users to activate password protected access to messages in their directories. Log files in the user’s directory capture a record of user actions on messages (for data analysis), messages recently removed, and any errors encountered while extracting messages (for maintenance purposes). In addition to user-specific data, separate directories on the web server receive updated audio and text files from information services such as hourly newscasts, weather and traffic reports. Currently all *VoiceCues* created by a user are stored in individual user directories and shared *VoiceCues* can be placed in a central location for all users. Overall, the objective is to provide a robust, secure and distributed server environment for multiple users and information services. In the future, the process of creating directories and default preferences for new users will be automated on the web server.

6.1.2 Filtering and Archiving via the Message Server

All incoming email and voice messages are processed by a *Message Server* activated by the user's *.forward* file. The *.forward* file specifies the user name and the host machine where the user's Nomadic Client is currently running. This allows the message server to notify the appropriate Client when new messages arrive. The *Message Server* parses the header for each incoming message to extract information such as date, time, sender, subject and the message body. Voice messages are converted from *.snd* format on the SUNs to *.wav* files for access on PCs. Static and dynamic filtering allows the system to set a priority level for each incoming message. Static filtering rules are specified by the user in her *.procmailrc* file on the server. These rules determine the priority of the message by comparing the sender and subject with pre-defined user preferences. In addition, content-based email filtering is integrated in *Nomadic Radio* using CLUES, a filtering and prioritization system developed in the Speech Interface Group [Marx 95, 96b]. CLUES determines the timely nature of messages by finding a correlation between a user's calendar, rolodex, to-do list, as well as a record of outgoing messages and phone calls. These messages are marked as "timely" when saved and subsequently notified to *Nomadic Clients*.

```
Date: Mon, 04 May 98 12:24:07 EDT
From: Operator <root@media.mit.edu>
Subject: Voice message from Geek 253-5156
Location: vmail.1.000323
Priority: most important
Read: true
Duration: 16
#####
Date: Mon, 04 May 1998 14:54:51 +0530
From: Rasil Ahuja <razzdazz@vxindia>
Subject: did you send your mom a card for mommy's day?
Location: msg.894273858
Priority: timely
Read: false
Duration: 2
#####
```

Figure 6.2: An example of message-related information updated in the user's *message file* on the web-server. This is read by the *Nomadic Clients* to download messages during startup or upon notification while running.

Processed copies of all messages are stored in the user's directories on the web server. A master *message file* for each user (see figure 6.2) is updated with header information, location, priority and the read status of each message. *Nomadic Clients* read the user's *message file* to download email and voice messages when they are first initiated and for notifications while running. A server-based implementation also allows the *Nomadic Clients* to remotely manage messages and action data. This is necessary since Java applets running in browsers are generally restricted to read-only access of data on local and remote machines. Clients communicate with the *Message Server* via sockets to send requests for saving and removing messages and actions on the web server. This provides an added layer of reliability and security for data management by only permitting authenticated clients to modify the user's message file and also preventing the message file from getting erroneously corrupted or overwritten.

6.1.3 Hourly updates from the News Server

A satellite dish receives hourly newscasts from ABC Radio (5 minute long audio summaries). The *News Server*, running on a SUN Sparcstation, automatically converts and archives these newscasts on the web server. *Nomadic Clients* that subscribe to ABC News receive an hourly notification from the *News Server*, to download the appropriate news audio file.

6.1.4 Notifications via the Radio Server

A *Radio Server*, a Java *application*, running on the wearable PC receives notifications for incoming email, voice and news messages. These are then forwarded to the Nomadic Client running on the same machine. This approach was utilized to handle the security restrictions on web browsers for accessing data on remote machines.

```
Sat May 09 15:59:07 Eastern Daylight Time 1998
k2.media.mit.edu > msgs/email.894744043

Sat May 09 16:06:54 Eastern Daylight Time 1998
marcy.media.mit.edu > msgs/news.894744472
```

Figure 6.3: Notifications for email and news (from different machines) received by the *Radio Server* and forwarded to the Nomadic Client on the wearable PC.

6.1.5 Location awareness via the Position Server

A *Position Server*, developed by John Holmes in the Speech Group, provides position and activity data. It keeps track of the user's position throughout the day using a distributed IR location system at the Media Lab, called the *Locust Swarm* [Starner97b]. Location awareness requires integration of IR-based receivers on the user's wearable PC. The *Position Server* also polls user activity data (on desktop machines) from the *Activity Server* [Manandhar91]. In the future, this data will be used to provide a location context for message notifications and asynchronous communication between nomadic users (see the future work section).

6.2 Wired Wearable vs. Distributed Wireless Operation

The *Nomadic Clients* have been developed in Java and use sockets for communication with remote information services as well as for integration with speech synthesis, recognition and audio monitoring modules. These modules can be distributed on networked machines and run remotely, based on the needs of the specific wearable configuration. A multi-threaded design of the *Nomadic Client* ensures fluid interaction for users, by performing all asynchronous operations as background parallel processes. Such processes include separate threads launched for downloading text and audio messages, playing and scanning audio streams, synthetic speech feedback, voice recognition and audio monitoring. Hence, the user experiences no slow-down while these processes are running, which is critical for real-time operation on an audio-only interface. Threads are utilized to synchronize the timing of auditory cues as well as synthetic speech and spatial audio streams for a coherent and well-paced presentation.

6.2.1 Wired Wearable Audio

Several alternatives were considered as a wearable computing platform for *Nomadic Radio*, based on in-house hardware and recently introduced commercial systems. An effort by lead by Thad Starner at the Media Lab created the *Lizzy*[18] platform [Starner97c], which is predominantly used by most wearable users here. The *Lizzy* is generally assembled with a 486 or 586-based processor, 16 MB RAM and 1GB hard-drive (figure 6.4). Specialized PC-104 cards can be installed for audio and video I/O. The *Lizzy* supports a *Private Eye* heads up display (red monochrome 720x280 resolution) and has a 10 hour battery life. The *Lizzy* is generally configured with a Linux operating system (OS). *Nomadic Radio* requires the use of a Windows-based OS for PC-dependent interface services such as speech recognition and spatialized audio. In our experience, installing a Windows-based OS on the *Lizzy* did not provide a reliable solution for audio I/O and wireless networking. As a result, several commercial systems were considered.

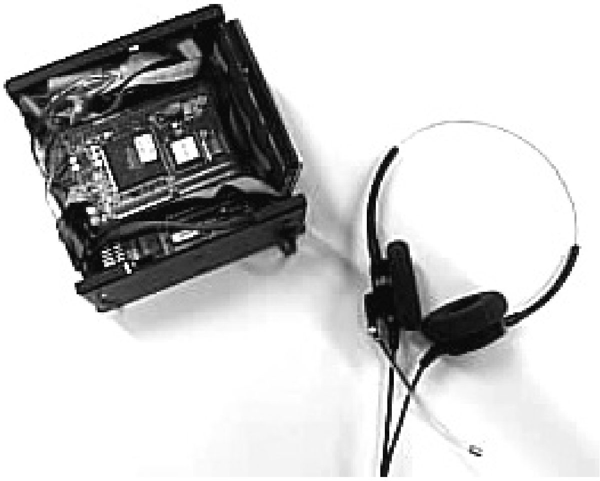


Figure 6.4: The *Lizzy* wearable platform developed at the MIT Media Lab, shown here configured with audio I/O for *Nomadic Radio*.

The main companies with commercial wearable products include InterVision, Phoenix Group Inc., Xybernaut (formerly CPSI), and ViA. A *VIA Wearable*[19] was acquired for evaluation in *Nomadic Radio* (see figure 6.5). It was configured with a 133 MHz 586 processor, 340 MB PC-Card hard-drive, 24 MB memory and 4 PC-Card slots. It uses a 7.2-volt NiMH rechargeable battery (2-3 hours duration). The VIA provided a lightweight, expandable and *wearable* (figure 6.5) alternative to the Lizzy and the Toshiba *Libretto* mini-notebook PC (evaluated earlier). Worn as a belt, the *ViA Wearable* has flexible sections which bend for comfort. The PC slots provide an easy mechanism to add networking, additional storage or services such as GPS (global positioning system). A "wired" configuration requires the *Nomadic Clients* and the speech and audio modules to operate on the wearable itself.

[18] *http://wearables.www.media.mit.edu/projects/wearables/lizzy/*

[19] *http://www.flexipc.com/Webpages/wearable.htm*

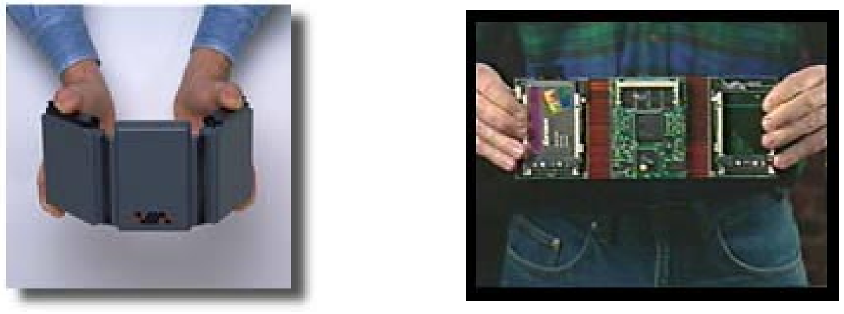


Figure 6.5: The *VIA Wearable* from Flexipc, shown here with the flexible circuit-board.

The different modules were initially operated on the *VIA Wearable* and the Toshiba *Libretto* PC. The memory and processing requirements of spatial audio, speech recognition and the Java-based clients, make their reliable operation challenging, on either platform. Running multiple interface services also requires independent audio channels. A software-based audio mixer is needed to manage audio from each service, however API's for speech recognition and spatial audio do not support independent audio control via a mixer. We found that audio from each service restricted others from utilizing the audio channel (even on full-duplex audio boards). As a result, we considered an approach utilizing distributed PCs.

6.2.2 Distributed Wireless Wearable Audio

An alternative configuration allows distributed wireless operation where the clients run on standalone desktop PCs or wearables, but the speech and audio components run remotely on networked PCs. This minimizes the computing and memory requirements on the wearable. In addition, this allows independent audio channels to be assigned for speech recognition, audio monitoring, speech synthesis and spatial audio, such that two or more systems can be simultaneously active. In the current architecture, wireless microphones and wireless stereo transmitters are utilized for delivering full-duplex audio from two networked PCs, each dedicated to different audio interface services. Wireless tactile input is provided by a numeric keypad connected to a networked *Libretto* mini-notebook PC.

The *Nomadic Clients* automatically detect the presence of local or remote speech/audio modules and switch their functionality accordingly. If local modules are detected, the system *interlaces* (activates one while deactivating others) the spatial audio playback with speech recognition and synthesis. In local speech mode, recognition must be activated on a push-to-talk basis. For remote modules, the system operates to provide continuous speech recognition and audio playback. This allows the user to *barge-in* or interrupt speech/audio playback with spoken commands. When a new message arrives and the user activity is determined to be low, audio monitoring is switched on to detect the conversation level (speech recognition is deactivated if operating on the same PC). Such an architecture, provides a flexible and modular mechanism to scale the *quality of services* based on the local and distributed processing available and the independent audio channels supported in a range of wired wearable and wireless audio platforms.

7. Conclusions

This thesis explored the design and development of a wearable audio platform, *Nomadic Radio*, for nomadic access to unified messaging. Interface techniques were designed for simultaneous and peripheral listening, spoken feedback, and navigation via voice and tactile interaction. The thesis addressed a crucial problem with current nomadic devices by demonstrating *contextual notification* in *Nomadic Radio*, i.e. using context to minimize interruption to users while providing timely notification of incoming messages.

This final chapter discusses the contributions of this work as well as insights gained and issues uncovered from iterative design in *Nomadic Radio*. The chapter will suggest interface enhancements to address these issues and propose open research areas for future work in *wearable audio computing*.

7.1 Contributions

Nomadic Radio is distinguished from previous messaging systems and mobile audio devices in several key areas. These techniques have been developed within the framework of a wearable computing platform. However, they can be generalized and applied to a variety of nomadic interfaces.

- **Contextual Notification:** A critical aspect of a wearable messaging system is to minimize interruptions to the user, while delivering timely notifications. In *Nomadic Radio,* contextual cues such as user activity, conversation level, and message priority affect the level of notification provided. The user's actions during notification reinforce the model dynamically. Such a model can also be utilized in personal devices such as pagers and cell-phones to manage notifications.
- **Peripheral Awareness and Notification:** To avoid distracting the user with recurring speech feedback, a variety of subtle auditory cues convey operational events, mode transitions, and priorities of incoming messages. A novel approach, *VoiceCues*, identifies the sender of an email by playing a unique audio signature. Continuous ambient cues provide an awareness of the operational state of the system and the status of messages being downloaded. In this manner, *Nomadic Radio* can operate entirely as a light-weight interface with minimal user interaction.
- **Dynamic Operational Modes:** On a wearable with limited resources, a dynamic state model allows the system to transition to states with minimum auditory interruption and consequently lower resource usage. These transitions occur gradually over time if the system detects no user activity, incoming messages with lower priority, or high conversation in the environment.
- **Spatial Listening Techniques:** A clock-like spatial layout of messages encodes their arrival time, to support recall and segregation for simultaneous audio streams. Spatial *foregrounding* of audio messages eases focus of attention by fading messages gracefully. The *scanning algorithm*

interlaces several audio streams in parallel, allowing the listener to hear an overall preview of the message space in an efficient manner.

- **Coherent Audio and Visual Displays**: A coherent metaphor for representing the message space in both the visual and auditory domains allows a familiar transition between desktop and nomadic (audio-only) modes of use. Messages attributes are translated into parallel auditory and visual representations to facilitate easier learning between the two displays. Coherent views enable rapid transitions between displays, i.e. if the audio wearable has an auxiliary visual display, it must replicate the user’s message space in a coherent manner to be effective on-demand.
- **Unified Access:** *Nomadic Radio* provides a *modeless* interface that allows users to interact with heterogeneous audio and text-based information in a similar manner. *Dynamic views* allow users to selectively filter their messages based on specific attributes. A familiar and consistent interface allows ease of navigation and browsing even as users subscribe to new information sources.
- **Rich Audio Interface via Distributed Services**: To overcome the limitations of current wearable computers, interface services such as spatialized audio, continuous speech recognition, audio monitoring, and tactile input were distributed between local and remote networked machines. By seamlessly merging these services on a wireless wearable, this approach provides a natural and fluid audio environment (with synthetic speech and simultaneous audio sources), while permitting the user to *barge-in* at any time via spoken commands or tactile input.

7.2 Insights and Issues

Design of the interface and development of the software functionality in *Nomadic Radio* produced several insights regarding effective interface strategies for wearable access to information.

- An initial goal of the project was to place all functionality and interface services on the wearable platform. Due to the memory, power and processing constraints of existing wearable computers, it became clear that an effective demonstration would require scaling down most audio interface techniques. An alternative approach allowed distribution of services on remote machines using the wireless-LAN infrastructure. This not only minimized the load on the wearable, but also allowed a robust demonstration of the interface techniques in an inter-office environment (where a distributed audio interface solution is feasible). Hence, although an integrated wearable solution is desirable, a distributed solution allows rapid prototyping and early evaluation of such techniques.
- In an audio-only interface, navigation and browsing must be carefully designed to provide concise and unambiguous feedback and allow natural and direct interaction. In a non-visual interface start-up conditions are important and hence, the user must be notified about the state of her message space on startup. A navigational home base, i.e. "go to my messages", allows users to easily situate themselves when lost or confused. Navigational feedback via auditory cues and spoken prompts is

provided while browsing messages and switching between views. Due to the temporal nature of listening to messages, timing and synchronization of audio playback, synthetic speech and auditory cues are critical for coherent display of messages. During scanning, message target windows are necessary to select the right message. Like *VoiceNotes*, the user's position in the navigation space is maintained at all times and moving past beginning or end of messages in a view anchors the user at the end points.

- An early design goal was to provide "hands-free" interaction using spoken commands only. Both push-to-talk and continuous speech techniques were explored. Continuous speech seems more natural but is disruptive when it picks up normal conversations. Although push-to-talk allows more control, it quickly becomes tedious. Hence, both techniques are allowed in *Nomadic Radio* to support their unique affordances, i.e. continuous speech provides a natural interface whereas push-to-talk allows the user to speak in noisy situations. Tactile interaction was added to the system primarily for testing purposes, however, it has now been adopted as a key interface element in conjunction with spoken commands for more efficient interaction.
- A spatial audio interface for messaging would be useful only if it provided a scaleable means for positioning all incoming messages. A time-to-space mapping allows a coherent representation that aids recall. However, differences in the user's abilities for listening to spatialized audio and level of noise in the environment reduce such benefits. In these situations, spatial audio primarily serves to segregate simultaneous audio sources and enhances foregrounding and scanning of messages.
- The notification model developed for managing interruptions to the listener evolved over several iterations. Initially, weights were assigned as default values, whereas later an interactive display allowed users to visualize the pattern of notifications over time and adjust the model accordingly. In a recent iteration, logarithmic scaling was found to provide a faster decay in reducing undesirable notifications while weighing messages with higher priorities at a greater level.
- The use of distinct auditory cues for notification in conjunction with a computed latency in delivering the notification provides users sufficient time to decide if they wish to hear the actual message. This reduces cognitive overhead from undesirable messages and pre-cues the user's attention more effectively for timely messages.
- Positive and negative reinforcement based on the user's actions adjusts the current thresholds of the notification model for subsequent messages. However, effective long term reinforcement learning requires that the contextual parameters (activity, conversation and priority levels) be captured for each user action. Hence, future messages arriving in a similar contextual situation should be assigned an appropriate notification level, based on prior listening patterns of the user. There are several problems and issues in formal reinforcement learning techniques that must be taken into account [Kaelbling96] in future work.

7.3 Future Work

7.3.1 Audio Interface Enhancements

Several improvements can be made to the existing audio interface by utilizing effective tools as follows:

- A natural sounding synthetic voice can have a significant impact on the usability and tolerability of spoken feedback and text message delivery. We are considering use of the *Laureate* synthetic speech system from British Telecom as a alternative to the AT&T's *FlexTalk* system.
- Message summaries for email are currently provided in *Nomadic Radio* as the first 100 characters of the message. This usually captures the relevant context of the message, yet lacks an overview. We hope to integrate a tool such as *ProSum* from British Telecom, which uses natural language parsing to summarize arbitrarily large documents.
- In *Nomadic Radio*, messages can be played at faster rates by increasing the pitch, but the resulting audio is less comprehensible. There are a range of techniques for time-compressing speech without modifying the pitch, based on SOLA (synchronized overlap add method) [Arons92]. In future versions of *Nomadic Radio*, such time-compression would provide better message previews.
- A mechanism for replying to email and voice messages requires voice recording, speech based dictation, or use of pre-composed email replies triggered by tactile or spoken commands. These approaches must be implemented and evaluated in the future.
- An extension of *Nomadic Radio* would be to support recording for capturing spontaneous conversations, lectures or interviews. This is not unlike *VoiceNotes*, however, such recordings could be indexed with contextually relevant cues such as time, location, and identity of speakers.
- Ideally before initiating, the system should verify the identify of the person via a user name and password which is spoken or typed on a numeric keypad for privacy. Alternatively, *Speaker Verification* on a recorded phrase of the user's voice could identify the user unobtrusively.

7.3.2 Haptic Feedback

Notification of events can be enhanced by haptic feedback[20] delivered via transducers mounted under the *SoundBeam Neckset.* This allows subtle physical awareness of incoming messages on the user's shoulder. However, rather than the binary vibrations provided by most pagers, variable feedback allows feedback proportional to the priority of the notification. It is conceivable that a number of small transducers may permit unique physical signatures, perceived by a user as identifying the sender of a message (similar to *VoiceCues*). Haptic feedback can provide an awareness of the dynamic status of messages being downloaded (analogous to ambient audio cues). A haptic modality would easily complement audio presentation in noisy environments and potentially reinforce spatial audio perception.

[20] *These ideas are based on personal discussions with Hiroshi Ishii*

7.3.3 Integration with Position Server for Location Awareness

Location awareness provides a means for evaluating the relevance of a user's current location relative to the messages presented. In classrooms and meetings, the user may expect minimum interruption. However, in specific meeting rooms, messages related to the meeting context should be prioritized. In addition, calendar reminders about meetings can be sensitive to the user's current location, reminding the user when she is expected in a certain location (if not sensed to be there already). Location awareness can be added to *Nomadic Radio* by integrating IR (Infrared) receivers on the wearable device. These IR receivers can provide positioning data to a *Position Server*, being developed by John Holmes in the Speech Group, using the Locust Swarm [Starner97b], a distributed IR location system at the MIT Media Lab.

The *Position Server* keeps track of the user's location throughout the day and polls user activity data from the *Activity Server* [Manandhar91]. *RadioSpace* is a Java-based client, developed by Natalia Marmasse in the Speech Interface Group, that communicates with the *Position Server* to provide a visual display of user activity as well as auditory cues. *Nomadic Radio* will utilize the same protocols as *RadioSpace* to communicate with the *Position Server*. This will provide an additional context that can influence the system's operational modes, level of message notification and scale the amount of speech feedback delivered.

Helping users gain an awareness of their colleagues for effective communication using a wearable audio platform is ideal. Spatial auditory awareness cues will indicate when people login, move to other rooms or logout, as detected via the *Position Server* that keeps track of a selected user community. The listener asks "who's there" and the system speaks the names of people currently available or use auditory cues to indicate their presence with varying levels of privacy. The listener can also track one or more users, such that their activity is continuously conveyed to her for a period of time.

7.3.4 Techniques for Awareness and Communication

Nomadic users can allow others to listen into their conversations to know if they can be interrupted, via a garbled audio recording, i.e. an audio stream where the voice content is encrypted while the identity of speakers is recognizable. The *GarblePhone*, an application currently being developed in the Speech Interface Group, will be incorporated within *Nomadic Radio* in the future. A comprehensive communication system should enable users to send asynchronous voice messages or initiate direct voice conversations with others, depending on their current level of interruption. The message could be broadcast to all users, and others should be able to asynchronously reply at their own pace. This form of communication is not unlike textual MUDs (multi-user domains) or Zephyr (a real-time text-based messaging service at MIT), with the added spontaneity and intonational properties of voice.

We are investigating the use of UDP (*User Datagram Protocol*, a connection-less network transport protocol) for synchronous voice communication in *Nomadic Radio*, using public-domain or commercially available Internet telephony APIs. Several modes of awareness and communication can be provided:

Abstract Awareness: spatial auditory cues indicating activity of users tracked by the *Position Server.*

Garbled Awareness: playback of garbled audio via *GablePhone* using UDP network protocol.

Asynchronous Voice Messaging: audio recording and playback on client with server messaging.

Synchronous Voice Communication:use of Internet telephony API for reliable half-duplex audio.

	Incoming	Outgoing
Asynchronous	• *Auditory cues for message notification* • *Messages downloaded periodically like email, voice mail, and news.*	• *Voice recording sent to person* • *Captured audio archived on remote web server*
Synchronous	• *Auditory cues for user activity.* • *Garbled audio awareness for privacy* • *Voice communication from caller*	• *Voice communication with 3rd party* • *Captured audio broadcast to multiple sites (via IP Multicasting)*

Table 7.1: Taxonomy of awareness, messaging and communication based on incoming or outgoing modes and asynchronous or synchronous transmission

7.3.5 Interaction Protocols for Communication Transitions

Currently *Nomadic Radio* provides only incoming-asynchronous messaging. In the near future, we plan to integrate incoming and outgoing-synchronous communication. Several interface issues must be considered for notification and to allow seamless transitions between different communication modes.

1. What form of notification will be provided for asynchronous messaging vs. synchronous communication? For example, how would a notification for a voice mail differ from a request by a caller to initiate a voice conversation with the user? How will this notification indicate the urgency level of a message or communication from a caller?

2. How will a user request garbled audio awareness of another user's auditory channel? How will the user be notified when another party wishes to hear their garbled audio stream? How should privacy levels be set for specific users and groups?

3. In what way should users be notified (or made continuously aware) that they have an open audio channel or garbled audio filter with another caller or audio multicast to several remote sites?

4. Some users may wish to filter their messaging/communication such that they can hear the party leave a voice message, while having the ability to intercept the caller to initiate a voice conversation. How should such a transaction be handled such that the caller is notified that the user has suddenly become available? How would the user place an existing party on hold or easily interleave between them?

5. What interface protocol must be used to notify or interrupt a user who is currently conversing with a caller, when a 2nd caller wishes to initiate communication with the user? In such a scenario, the user could be notified via audio cues or synthesized speech regarding the caller and the 2nd caller

could receive some speech/audio notification that the user is busy. Alternatively, the caller could also receive a synchronous/asynchronous garbled audio of the conversation if allowed by the user.

This indicates a somewhat complex set of transaction protocols and interface issues that must be handled in an unobtrusive manner without confusing either the user or the caller while maintaining a desirable level of privacy and notification. Related studies in audio only communication environments offer some insights [Hindus96][Watts96]. These issues are particularly challenging since notifications and complex transactions between users must be provided via a speech and audio-only modality.

7.3.6 Environmental Audio Classification

Our initial efforts have focused on discriminating speech from ambient noise in the environment. In the future, a rich classification of background sounds will help establish user context in offices, classrooms, trains, cafes, and the outdoors. The techniques for feature extraction and classification, could be ported to run efficiently on a wearable platform with continuous audio buffering of environmental sounds. The system could be continuously updated with new training data in different locations and adaptively build more robust models of granular sound classes from several environments over time. This is particularly beneficial when a user's exact location can not be determined due to lack of location sensing infrastructure in the environment. However, cues based on position (via GPS and IR), time of day, and user/domain knowledge can greatly aid in reducing ambiguities in sound classification.

7.3.7 Evaluation

Several key aspects of the auditory interface and adaptive notification can be evaluated in the future, based on regular usage of *Nomadic Radio* in both stationary (office usage) and nomadic environments (wearable usage). Such issues can be evaluated using common usability criteria, such as, learnability for novice users and ease of use for users with some prior experience.

- How well can users distinguish auditory cues for awareness, status and priority of messages?
- How well do users comprehend more than one spatial audio stream during scanning?
- To what extent do users utilize voice or tactile input for navigation and browsing messages?
- The performance of the notification model must be determined by considering the rate of errors and improvement over time as more data is acquired. An error is defined as the difference in predicted presentation vs. one desired by the user. Since the system is continuously adapting to the user's choices within the context of changing parameters in the environment, we expect that the performance of the model will vary widely during the day.

For short-term evaluation, 2-3 subjects would be asked to try a series of interaction tasks for notification and browsing (using a sample set of messages), with a brief training time. Longer-term field usage by myself and speech group students (over 2-3 months), allows iterative refinement and qualitative assessment. Such usage will especially help evaluate the adaptive notification strategies over time.

References

[Ahlberg94] Ahlberg, C. and B. Schneiderman. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Display. *Proceedings of CHI ‘94*. pp. 313-317.

[Arnaud95] Arnaud, Nicolas. Classification of Sound Textures. M.S. Thesis, Media Arts and Sciences, MIT Media Lab, September 1995.

[Arons92] Arons, Barry. A Review of the Cocktail Party Effect. *Journal of American Voice I/O Society*, Vol. 12, July 1992.

[Arons93] Arons, Barry. SpeechSkimmer: Interactively Skimming Recorded Speech. *Proceedings of UIST’ 93*, November 1993.

[AT&T97] WATSON for Windows, Version 2.1, System Developer’s Guide. AT&T, 1997. *http://www.att.com/aspg/blasr.html*

[Bederson96] Bederson, Benjamin B. Audio Augmented Reality: A Prototype Automated Tour Guide. *Proceedings of CHI ’95*, May 1996, pp. 210-211.

[Blauert83] Blauert, J. *Spatial Hearing*, trans. John S. Allen. MIT Press, 1983.

[Bregman90] Bregman, Albert S. *Auditory Scene Analysis: The Perceptual Organization of Sound.* MIT Press, 1990.

[Cahn90] Cahn, Janet E. The Generation of Affect in Synthesized Speech. *Journal of the American Voice I/O Society*, July 1990.

[Chalfonte91] Chalfonte, B.L., Fish, R.S. and Kraut, R.E. Expressive richness: A comparison of speech and text as media for revision. *Proceedings of CHI '91*, pp. 21-26. ACM, 1991.

[Clarkson98] Clarkson, Brian and Alex Pentland. Extracting Context from Environmental Audio. Submitted to the *International Symposium on Wearable Computing,* IEEE, 1998.

[Cohen94] Cohen, J. Monitoring background activities. *Auditory Display: Sonification, Audification, and Auditory Interfaces*. Reading, MA: Addison-Wesley, 1994.

[Conaill95] Conaill, O’ Brid and David Frohlich. Timespace in the Workplace: Dealing with Interruptions. *Proceedings of CHI `95*, 1995.

[Gardner95] Gardner, W. G., and Martin, K. D. HRTF measurements of a KEMAR. *Journal of the Acoustical Society of America*, 97 (6), 1995, pp. 3907-3908.

[Gaver89] Gaver, W.W. The Sonic Finder: An interface that uses auditory icons. *Human Computer Interaction*, 4:67-94, 1989.

[Gaver91] Gaver, W.W., R. B. Smith, T. O’Shea. Effective Sounds in Complex Systems: The ARKola Simulation. *Proceedings of CHI '91*, April 28-May 2, 1991.

[Handel89] Handel, S. *Listening: An Introduction to the Perception of Auditory Events.* MIT Press, 1989.

[Hanson96] Hansen, Brian, David G. Novick, Stephen Sutton. Systematic Design of Spoken Prompts. *Proceedings of CHI '96*, April1996, pp. 157-164.

[Hayes83] Hayes, P. and D. Reddy. Steps towards graceful interaction in spoken and written man-machine communication. *International Journal of Man Machine Studies*, 19, pp. 231-284, 1983.

[Hindus96] Hindus, Debby, Mark S. Ackerman, Scott Mainwaring, and Brian Starr. Thunderwire: A Field Study of an Audio-Only Media Space. *Proceedings of CSCW'96*, pp. 238-247. November 1996.

[Horvitz95] Horvitz, Eric and Jed Lengyel. Perception, Attention, and Resources: A Decision-Theoretic Approach to Graphics Rendering. *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI'97),* Providence, RI, Aug. 1-3, 1997, pp. 238-249.

[Horvitz97] Horvitz, Eric and Matthew Barry. Display of Information for Time-Critical Decision Making. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI'95),* Montreal, August 1995.

[Hudson96] Hudson, Scott E. and Ian Smith. Electronic Mail Previews Using Non-Speech Audio. *Proceedings of CHI '96*, April 1996, pp. 237-238.

[Intel97] Intel Realistic Sound Experience, 3D RSX SDK Documentation, Intel Corporation, 1997. *http://developer.intel.com/ial/rsx/doc.htm*

[Jebara98] Jebara, Tony, Bernt Schiele, Nuria Oliver, and Alex Pentland. DyPERS: A Dynamic and Personal Enhanced Reality System. Submitted to the *International Symposium on Wearable Computing*, IEEE, 1998.

[Kaelbling96] Kaelbling, Leslie Pack and Michael L. Littman. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research* , vol. 4, 1996, pp. 237-285.

[Kendall95] Kendall, Gary S. A 3-D Sound Primer: Directional Hearing and Stereo Reproduction. *Computer Music Journal*, 19:4, pp.23-46, Winter 1995, MIT Press.

[Klatt87] Klatt, D.H. Review of text-to-speech conversion for English. *Journal of the Acoustic Society of America*, 82(3): 737-793, 1987.

[Kobayashi97] Kobayashi, Minoru and Chris Schmandt. Dynamic Soundscape: Mapping Time to Space for Audio Browsing. *Proceedings of CHI '97*, March 1997.

[Luce83] Luce, P.A., T.C. Feustel and D.B. Pisoni. Capacity demands in short-term memory for synthetic and natural speech. *Human Factors*, 25(1):17-32, 1983.

[Ly94] Ly, Eric and Chris Schmandt. Chatter: A Conversational Learning Speech Interface. *Proceedings of AAAI'94 Spring Symposium on Multi-Media Multi-Modal Systems*, Stanford, CA, March 1994.

[Maes94] Maes, Pattie. Agents that Reduce Work and Information Overload. *Communications of the ACM*, July 1994, Vol.34, No.7, pp. 31-41.

[Maher97] Maher, Brenden. Navigating a Spatialized Speech Environment through Simultaneous Listening and Tangible Interactions. M.S. Thesis, Media Arts and Sciences, MIT Media Lab, Fall 1997.

[Manandhar91] Manandhar, Sanjay Activity Server: A Model for Everyday Office Activities. M.S. Thesis, Media Arts and Sciences, MIT Media Lab, June1991.

[Martin89] Martin, G.L. The utility of speech input in user interfaces. *International Journal of Man Machine Studies*, 30:355-375, 1989.

[Marx95] Marx, Matthew. Toward Effective Conversational Messaging. M.S. Thesis, Media Arts and Sciences, MIT Media Lab, June 1995.

[Marx96a] Marx, Matthew and Chris Schmandt. MailCall: Message Presentation and Navigation in a Non-visual Environment. *Proceedings of CHI ’96*, pp. 165-172, April 1996.

[Marx96b] Marx, Matthew and Chris Schmandt. CLUES: Dynamic Personalized Message Filtering. *Proceedings of CSCW ‘96*, pp. 113-121, November 1996.

[Muller90] Muller, M. and J. Daniel. Toward a definition of voice documents. *Proceedings of COIS '90,* pp. 174-182, ACM, 1990.

[Mullins96] Mullins, Atty T. AudioStreamer: Leveraging The Cocktail Party Effect for Efficient Listening. M.S. Thesis, Media Arts and Sciences, MIT Media Lab, February 1996.

[Mynatt98] Mynatt, E.D., Back, M., Want, R. Baer, M., and Ellis J.B. Designing Audio Aura. *Proceedings of CHI ’98,* April 1998.

[Norman94] Norman, Donald A. "The Power of Representation” in *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine.* Addison-Wesley Pub Co. May 1994, pp. 43-75.

[Nortel94] *SoundBeam* Design Qualification. Proprietary Report TL940041, Nortel, September 1994.

[Papp96] Papp, Albert and Meera M. Blattner, Dynamic Presentation of Asynchronous Auditory Output, *Proceedings of ACM Multimedia’96*, November 1996, pp.109-116.

[Pashler98] Pashler, Harold E. *The Psychology of Attention*. MIT Press, 1998.

[Rekimoto95] Rekimoto, Jun and Katashi Nagao. The World through the Computer: Computer Augmented Interaction with Real World Environments. *Proceedings of UIST ''95*, November 14-17, 1995, pp. 29-38.

[Rhodes97] Rhodes, Bradley J. The Wearable Remembrance Agent: a system for augmented memory. *Proceedings of the International Symposium on Wearable Computing,* IEEE, October 1997, pp. 123-128.

[Roy95] Roy, Deb K. NewsComm: A Hand-Held Device for Interactive Access to Structured Audio. M.S. Thesis, Media Arts and Sciences, MIT Media Lab, June 1995.

[Roy96] Roy, Deb K. and Chris Schmandt. NewsComm: A Hand-Held Interface for Interactive Access to Structured Audio. *Proceedings of CHI ’96*, April 1996, pp. 173-180.

[Roy97] Roy, Deb K., Nitin Sawhney, Chris Schmandt, and Alex Pentland. Wearable Audio Computing: A Survey of Interaction Techniques *Perceptual Computing Technical Report No. 434*, MIT Media Lab, April 1997.

[Rudnicky96] Rudnicky, Alexander, Stephen Reed and Eric Thayer. SpeechWear: A mobile speech system. *Proceedings of ICSLP '96*, 1996.

[Sawhney97] Sawhney, Nitin. Situational Awareness from Environmental Sounds. Project Report for Pattie Maes, MIT Media Lab, June 1997.
http://www.media.mit.edu/~nitin/papers/Env_Snds/EnvSnds.html

[Schmandt93] Schmandt, Chris. Phoneshell: The Telephone as a Computer Terminal. *Proceedings of ACM Multimedia '93*, pp. 373-382, New York, Aug 1993.

[Schmandt94a] Schmandt, Chris. Multimedia Nomadic Services on Today's Hardware. *IEEE Network*, September/October 1994, pp12-21.

[Schmandt94b] Schmandt, Chris. *Voice Communication with Computers: Conversational Systems*, Van Nostrand Reinhold, New York, 1994.

[Schmandt95] Schmandt, Chris and Atty Mullins. AudioStreamer: Exploiting Simultaneity for Listening. *Proceedings of CHI 95*, pp. 218-219, May 1995.

[Starner97a] Starner, Thad, Steve Mann, Bradley Rhodes, Jeffery Levine, Jennifer Healey, Dana Kirsch, Rosalind Picard, and Alex Pentland, Augmented Reality through Wearable Computing. *Presence*, Vol. 6, No. 4, August 1997, pp. 386-398.

[Starner97b] Starner, Thad and Dana Kirsch. The locust swarm: An environmentally-powered, networkless location and messaging system. *Proceedings of the International Symposium on Wearable Computing,* IEEE, October 1997.

[Starner97c] Starner, T. (1997). Lizzy Wearable Computer Assembly Instructions. MIT Media Laboratory. *http://wearables.www.media.mit.edu/projects/wearables/*

[Stifelman92] Stifelman, Lisa J. VoiceNotes: An Application for a Voice-Controlled Hand-Held Computer. M.S. Thesis, Media Arts and Sciences, MIT Media Lab, May 1992.

[Stifelman93] Stifelman, Lisa J., Barry Arons, Chris Schmandt, and Eric A. Hulteen. VoiceNotes: A Speech Interface for Hand Held Voice Notetaker. *Proceedings of INTERCHI '93*, New York, April 1993.

[Stifelman94] Stifelman, Lisa J. The Cocktail Party Effect in Auditory Interfaces: A Study of Simultaneous Presentation. MIT Media Lab Technical Report, September 1994.

[Stifelman96] Stifelman, Lisa J. Augmenting Real-World Objects: A Paper-Based Audio Notebook. *Proceedings of CHI '96*, April 1996.

[Voor65] Voor, J.B. and J.M. Miller. The effect of practice upon the comprehension of time-compressed speech. *Speech Monographs*, 32(452-455), 1965.

[Wallach40] Wallach, H. 1940. The Role of Head Movements and Vestibular and Visual Cues in Sound Localization. *Journal of Experimental Psychology*, 27(4):339-368.

[Watts96] Watts, Jennifer C., David D. Woods, James M. Corban, and Emily S. Patterson. Voice Loops as Cooperative Aids in Space Shuttle Mission Control. *Proceedings of CSCW'96*, pp. 48-247. November 1996.

[Wenzel92] Wenzel, E.M. Localization in virtual acoustic displays, *Presence*, 1, 80, 1992.

[Whittaker94] Whittaker, S. P. Hyland, and M. Wiley. Filochat: Handwritten Notes Provide Access to Recorded Conversations. *Proceedings of CHI '94*, pp. 271-279, 1994.

[Wilcox97] Wilcox, Lynn D., Bill N. Schilit, and Nitin Sawhney. Dynomite: A Dynamically Organized Ink and Audio Notebook. *Proceedings of CHI '97*, March 1997, pp. 186-193.

[Yankelovich94] Yankelovich, Nicole. Talking vs. Taking: Speech Access to Remote Computers. *Proceedings of CHI '94*, Boston, MA, April 24-28, 1994.