# Podscanning: Audio microcontent and synchronous communication for mobile devices

by

Patrick Sean Wheeler

B.A., Linguistics and Cognitive Science, Brandeis University (1991)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2010



Author_____

Program in Media Arts and Sciences
September 18, 2009

Certified by_____

Christopher Schmandt
Principal Research Scientist
MIT Media Laboratory
Thesis Supervisor

Accepted by_____

Prof. Deb Roy
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

**Podscanning: Audio microcontent and synchronous communication for mobile devices**

by

Patrick Sean Wheeler

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on September 18, 2009, in partial fulfillment of the
requirements for the degree of
Master of Science

## Abstract

Over the past decade, computationally powerful audio communication devices have become commonplace. Mobile devices have high storage capacity for digital audio, and smartphones or networked PDAs can be used to stream internet radio and download podcasts. However, applications that allow mobile devices to be used for synchronous communication (such as person-to-person audio interaction and listening to broadcast synchronous audio) are distinct from applications that allow stored audio playback.

To demonstrate the benefits of more fluidity in mobile user interfaces between asynchronous audio and synchronous audio playback, I have implemented a new user interface approach - audio scanning - that merges these data types in a single user interface.

A combined interface must solve two different research problems. Asynchronous messaging queues grow longer without constant user intervention. Synchronous audio, on the other hand, can be disruptive and transient. To address these problems, a timing heuristic is used in audio scanning to allow sporadic yet controlled delivery of organized audio bursts. To lessen the burden of user interaction with a graphical user interface on a mobile device, a single-button user interface allows control of audio presentation.

Two exploratory systems implementing an audio scanning interface are described, allowing comparison to alternate audio systems and approaches. The first implementation, Hearplay, demonstrates the utility of audio scanning in a social audio appliance, designed to be available as an always-on system. The second implementation, Hearwell, demonstrates a use of audio scanning on a mobile phone to help individuals achieve wellness goals. The design and utility of the scanning interface is critiqued for both implementations.

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

**Podscanning: Audio microcontent and synchronous communication for mobile devices**

by

Patrick Sean Wheeler

The following people served as readers for this thesis:

Thesis Reader_____

Stephen S. Intille
Research Scientist, House_n
MIT Department of Architecture

Thesis Reader_____

Patti Maes
Associate Professor of Media Technology
MIT Media Laboratory

## Acknowledgments

When I started graduate studies at the Media Lab, I could never have guessed how things would end, or more to the point, *when* things would end. John Lennon did say it best; life is what happens to you while you're busy making other plans. But the slight detours have only helped me to better appreciate the friendships and bonds that I formed in the Lab. This work is testament to these incredible people who have helped me along the way.

First and foremost, I am grateful to Chris Schmandt, both for giving me a place in the Speech Interface (+ Mobility!) Group these years ago as a research assistant, and for encouraging me to see this work completed. While you taught me so much at the Lab, you taught me even more these past few years about resilience and perseverance. You reminded me of the goal even when it seemed unattainable. Thank you for helping me see this through.

I'd like to thank my thesis readers, Stephen Intille and Pattie Maes, for helping me refine the presentation of my research with their whole-hearted attention. Thank you, Linda Peterson, for your help in clarifying what needed to be done and for your support in the process.

I am grateful to fellow Speech Group alumni who offered unwavering encouragement and assistance during my research. Natalia Marmasse, Stefan Marti, Vidya Lakshmipathy – even in stressful times I knew I could rely on your insights and advice. I miss you all.

My former France Telecom colleagues have also supported me in my return to the Lab. Thank you Pascal Chesnais, Lorrie LeJeune, Jason Sroka, Chris Roux, and Frank Bowman. Thanks as well to Lorraine Gray for helping me to see the eagle in the stone.

To all of those friends and colleagues who I first met at the Media Lab: to a person you continue to inspire me with your intelligence and drive. Jacky Mallett, I would include you in this widely scoped compliment, but you've taught me that the British tend to be suspicious of such effusive praise. So instead, thank you for the chocolate and for possibly figuring out what caused the global financial meltdown.

Finally, to Jimy, I think it must be your Irish tenacity, because you have never, ever given up. Thank you for being with me at every step of this journey.

# Contents

# Chapter 1

# Introduction

The mobile phone continues to rapidly evolve. According to the International Telecommunications Union, the number of mobile users surpassed 50% of the world's population rate in 2008 - a growth rate greater than that of the Internet [1, 2]. Mobile handsets themselves have become more powerful and feature-rich. Consumers change phones rapidly in comparison to other technologies such as laptop computers. The Semiconductor Industry Association reports that in 2006 the mobile phone replacement cycle dropped to an average of 18 months worldwide, but PC replacement remained steady at 3.5 years [3].

Aside from this evidence of rapid growth in the commercial market for mobile communications, there are reasons to believe that the adoption of mobile technology is causing deep shifts in how people interact with each other. The appearance of unanticipated consequences and usages is not new in the history of communications technology. For example, the introduction of the answering machine allowed the development of new and unanticipated forms of communication, making it possible to screen calls before answering, to play "phone tag", and even to distribute information in an outgoing message. More subtly, sociologists discovered that people using answering machines in both incoming and outgoing messages were conversational in leaving messages, even though the one-sided nature of communication was apparent [4].

Due to the rapid spread of mobile phones, people can stay in contact no matter where they are. With a mobile device, using a range of methods and applications, we can more efficiently organize the details and opportunities for social interaction regardless of distance.

Some researchers believe that this sort of mobile usage has created a more subtle effect in the nature of the interactions we have with others - whom we communicate with, how often, and why - among other details of daily life [5].

Mobility is also changing the expectations for interaction with information systems. Interacting with computers used to be limited to the desktop. With the growth of high speed mobile data (provided as 3G and 4G networks) it is no longer necessary to be at a desktop computer to gain access to remote data. Just as always-on connectivity has created distraction at the desktop, the availability of data through the mobile device has created more demands for user attention.

## 1.1 Influence of mobility on social interaction and access to information

Against this landscape of technological change, three particular ideas can already be emphasized to identify how mobility has altered the ways we access information and communicate with others. Though offered as axioms, these points will motivate the primary goal for this thesis - the design of a novel audio user interface to support new forms of mobile interaction.

- Research has shown that mobile phones allow people to communicate more sporadically, often with long pauses between communications while the user shifts attention to deal with other matters in daily life [6]. These communications are very brief but can also be very frequent. This effect has been called *micro-coordination* [7, 8].

- The attentional and interactional demands of interaction with mobile devices have compounded the existing attentional demands of interacting in the physical world, creating what some technologists have called a strategy of *continuous partial attention* in which the user never gives full attention to a device, but instead interacts sporadically and monitors for alerts and changes [9].

- The spread of the mobile network has created always-on connection with the electronic world. Now workers can do their jobs not only at the office, but in hotels, train stations, or the home. This creates new expectations for all those who interact in

the workplace on how, where, and when information can be used for workflow [10–12]. For example, when capturing information relevant to a task, workers will often capture tasks away from the desktop and are sensitive to the regularity of routines and schedules that span both the office and the personal world [13]. This kind of *fluid organization* can spill over from work practices to other personal roles to create an increasingly ubiquitous information environment.

No matter how individuals feel about these changes, it is unrealistic to expect that the disruptions caused by these shifts in interaction will be resolved by abandoning the benefits of mobility altogether. The rapid spread of mobile phones will continue to broaden the opportunities to use audio of all forms while on the move.

## 1.2 Drawbacks and benefits of audio interfaces on mobile devices

There are several difficulties that the use of audio produces in mobile situations. Some of these difficulties are caused by the inherent properties of audio as a communications medium. Without the benefit of headphones, audio playback at a normal volume can be heard by others nearby. It is difficult to avoid being overheard when speaking through mobile channels. Lengthy spoken interaction is a common nuisance in the workplace and other indoor spaces. Audio alerts can be an unwanted distraction. Listening to audio with headphones can isolate a user from desired social interaction. Audio is a transient medium; audio that is not stored cannot be retrieved, causing burden on human memory.

One reason that users have upgraded phones so rapidly is the functional advantage of text-capable, feature-rich smartphones over simple, voice-only mobile phones. Speaking on the phone can be disruptive in public. Using text messaging (including SMS, email, or mobile instant messaging), we can communicate privately with others wherever we have a network connection. A 2007 study showed that in the US, half of all mobile phone users used the phone for applications in addition to making calls, with text messaging taking the lead in usage with 36% of users using this function on their phone in the last month [14]. Data usage further will further increase as mobile web browsers improve and specialized applications

are developed. Bluetooth-equipped phones allow audio to be listened to privately, which makes the cell phone a private, less intrusive device when used in public.

In many mobile applications, however, voice and audio interaction is an asset. Despite the utility of SMS, email and mobile instant messaging, the phone call is still the easiest and most used communications method on the mobile device. Audio is an especially appropriate channel for information delivery whenever the users hands or eyes are occupied elsewhere. Audio notification can effectively deliver a message without distracting the visual attention of a user engaged in other tasks on the move, and without burdensome mobile text entry interfaces [15]. Audio is an emotive medium, and desirable in several forms when a text message is open to misinterpretation. Even the use of headphones can be desirable when the user wants to prevent unwanted social interaction.

## 1.3 My contribution

The increased computing power of the mobile device itself allows for exploratory design and evaluation of new ways of leveraging the utility of audio interfaces against their detractions on the mobile device. I present two systems in this thesis as contributions in this area. I precede a high-level description of both systems described in this thesis with a short illustrative use-case of each below.

### 1.3.1 Hearplay

> Setting his laptop bag and his mobile phone down on his desktop, Bob begins a day in his office. Even before his computer can boot up, he hears a chirp from his mobile and a stream of audio begins to play. He hears a short voicemail, a beep, and another voicemail. As soon as the voicemail ends, a song starts to play. Bob begins to get settled in as he listens to the music for a bit, but finally presses a button on the mobile's screen, and the song stops playing with a chirp.
>
> As he starts to focus on writing a document, Bob leaves his mobile on. Five minutes later, he hears the same stream of voicemail from his mobile. Bob knows he should return the calls, so he does nothing. He gets up, goes to get

a cup of coffee, and returns, just catching one of his co-workers on the mobile, asking if anyone has seen the boss. Fifteen minutes later he hears the same stream of voicemail. 'Maybe I don't need to return one of these calls today after all', he thinks. He picks up the mobile and presses a single button on the screen, enjoying the happy-sounding short chirp. Thirty minutes later he hears the lingering voicemail again. He knows if things are going well, he will hear this message less and less frequently as time goes on. Only three minutes later, though, he hears a chirp and a new audio message about a upcoming meeting from a colleague, before hearing the repetition of the first voicemail from this morning. The messages are happening more frequently with each sweep, he thinks. Bob knows it will be a busy day.

Hearplay was created as a general-purpose communication and shared annotation system that focuses on the utility of audio scanning for content aggregation in a shared group. At a high level, Hearplay is a system in which:

- users autonomously broadcast stored audio and synchronous audio to members within a shared broadcast space
- individual serial audio streams emerge from a combination of locally stored, downloaded, and synchronous audio sources
- repeated messages ensure that bursts of audio are heard, even without user interaction
- back-off in presentation of audio bursts ensure interruptions are short and managed

### 1.3.2 Hearwell

Gathering his things on the way out the door, Bob remembers to call his wife to see if he needs to stop at the grocery on the way home. He gets in his car, and places the call. Right after as he dials, he sees a dialogue screen appear on his mobile, *Would you like to listen to some wellness tips after your call? Yes / No.* Bob clicks yes, and the call proceeds. Bob needs to stop off and pick up milk on the way home.

> Bob hangs up, hears a beep and then a short burst of audio starts to play. *Stress can make you fat! You can take some easy steps..* 'Aha', Bob thinks, 'this might explain a few things'. He listens to the message all the way through. *Did you know eating late can be unhealthy? Studies show....* He listens to about 10 seconds, and presses *Skip this* on the screen to go to the next tip. He knows if he doesn't skip the audio, he will hear the message later, possibly even after his next call. Bob listens to the following burst, all about the health benefits of broccoli. 'I don't know about more broccoli', Bob thinks, 'but maybe I should go to the gym before dinner.'

Hearwell is an application of audio scanning to a mobile-supported wellness scenario. By listening to previously recorded audio content, users can remind themselves at opportune times to take action or reenforce learning. At a high level, Hearwell is a system in which:

- presentation of stored audio content is triggered by synchronous communication
- individual audio streams can be tailored from downloaded or locally stored content to meet personalized health goals
- repeated messages ensure that relevant information is remembered
- back-off in presentation of audio bursts assist the process of memorization

## 1.4 Structure of this thesis

Hearplay and Hearwell utilize a new user interface for mobile audio, which I call audio scanning[1], in an attempt to support new mobile usage patterns in a coordinated manner. These patterns include micro-coordination, continuous partial attention, and fluid organization of tasks. In this thesis, I apply the audio scanning interface in two use cases and evaluate its usage to identify future possibilities.

The structure of the thesis is outlined below:

[1]Audio scanning on a mobile device could also be called *podscanning*, especially when stored audio is used.

- To go more deeply into the need for new user interface metaphors to support mobility, chapter 2 will discuss related work in the areas of computer-supported collaborative work, ubiquitous and pervasive computing, and social media.

- Chapter 3 will outline the audio scanning interface, its source and its application to mobile audio appliances. The design goal is to enable microcoordination, fluid organization, and continuous partial attention to changing information sources. The audio scanning interface allows aggregation and playback of stored audio, interspersed by synchronous audio events. It presents information using an audio format or program that is consistent and understandable by the listener. After describing the general approach of audio scanning and its entailments, I describe two implementations of audio scanning in two separate use cases in the following chapters.

- In chapter 4, I describe the implementation of Hearplay in more detail, applying audio scanning to the problem of audio content aggregation and group communication.

- In chapter 5, I critique the design of Hearplay to inform the design of Hearwell, a special-purpose audio information appliance that supports health and wellness information seeking.

- Finally, I summarize how the audio scanning systems created as part of this work can support new usage patterns. I finish with a discussion on design implications of the audio scanning approach.

# Chapter 2

# Audio and Mobile Interfaces

The previous chapter raises the underlying observation that mobile devices have made people continuously available for communication from anyone, creating intermittent but critical demands on attention. The rapid spread in mobile technology and its social consequences have already changed how people initiate, engage, and break off communication and access information on these devices.

How then can designers better support the new forms of social and human-computer interaction on mobile devices? This chapter will investigate how previous work has focused on the application of audio to similar attentional demands in a wide range of applications, from media spaces to computer-supported collaborative work. This previous research is based on the insight that audio interfaces are well suited for use cases that require lightweight recognition of salient events, draw quick shifts in user attention, facilitate rapid interaction, or that require complex social consideration in follow-up as is discussed below.

## 2.1 A scenario

*Imagine that you are deeply engrossed in a phone call with a relative one evening. Even when you are fully engaged in the conversation, you also might hear the remote sound of the faucet running in the background, the clink of dish-ware and pans, and children playing in the background. Quite suddenly you hear a loud crack and the shattering sound of breaking glass. Do you continue to talk, or do you wait for your relative to respond, before going on*

*with the conversation?*

As this scenario shows, sensitivity and awareness to the audio environment is a familiar and expected part of everyday life. The quality of awareness in this scenario is marked by several general aspects which could also extend to the user experience of audio mobile interfaces. Though mundane, this scenario points towards certain user responses to the audio environment that are interesting for the design of audio communication systems:

- Ease of interpretation. We quite easily interpret audio information. We perceive subtle nuances of emotion in verbal communication, which helps us to plan an appropriate level of response. We quickly identify the importance of an unfamiliar sound even when the source is unseen.

- Background and salience. In everyday life, we are quick to notice an important sound, even when it does not carry our full attention. Much of this interpretation happens automatically, without need for conscious thought. On the contrary, it is the unusual sound that stands out and causes a rapid shift of attention.

- The influence of audio on behavior and interaction. The acoustic environment is all-surrounding and influences our interpretations of the physical environment in a direct manner. Audio is a key sensory medium for ambient awareness. Since social interaction is influenced by the shared physical environment, we may expect that the acoustic environment can influence the forms of social interaction in ways that might be exploited by the designer.

## 2.2 Audio interfaces to support continuous partial awareness

Immersive audio environments have been created to support distributed collaborative work, and interactive educational and artistic installations in primarily public or semi-public spaces. The goal of these systems is to support background awareness to deliver relevant information without requiring explicit user request or interaction. The design of these systems is directly relevant to support for continuous partial awareness in mobile user interfaces.

In this section, I will describe two relevant projects: *Audio Aura* and *A Walk in the Wired Woods*, and draw some observations for applying this research to mobile audio interfaces.

### 2.2.1 Audio Aura

Audio Aura is a workplace system to provide an opportunistic, spontaneous audio experience that relates information from the virtual world to users immersed in the physical environment [16]. The purpose of Audio Aura was to enrich office worker's background awareness of changes in the virtual environment, including email and calendar information. The hypothesis is that such background awareness can facilitate workplace coordination and interaction. Computer-mediated audio environments such as Audio Aura rely on the ability of computer systems to create an artificial sound environment to pull users to enter into symbolic interaction that is only indirectly connected to a physical environment. As the user moves throughout the workspace, Audio Aura captures events (whether a user leaves or exits a location), and represents this event using audio cues to other users.

Context information was captured by an infrared sensor/badge system that allowed AudioAura to track a user moving throughout the workplace. Based on these movements, an artificial audio environment was created. Audio was delivered via wireless headphones. The audio design of the system explored alternative sound ecologies - speech, music, or even sound effects, to create a non-intrusive environment for continuous partial awareness.

### 2.2.2 A Walk in the Wired Woods

A Walk in the Wired Woods used recorded audio to augment the experience in a physical gallery installation of woodland photographs [17]. As a visitor walked around the gallery, physical location was used to determine which sounds would enhance the experience, depending on the content of the photograph. Approaching a photo of a bird on a branch, for example, the visitor might hear birdsong. Walking around the installation, the user might hear music, spoken fragments, or other sounds depending on location. Similar kinds of immersive audio experience have become commercialized in various museums and audio guides.

A Walk in the Wired Woods identified the need for technical features to allow a greater potential degree of customization. The developers implemented an XML markup language to allow experience designers to specify audio effects and characteristics based on location including looping, fading, and mixing.

### 2.2.3 Lessons learned from immersive audio environments

- Immersive systems are primarily a form of context-driven human-computer interaction. These systems augment rather than replace social interaction, which is the key capability provided by the mobile.

- Two-way person-to-person interaction is not the central design goal of these systems. These systems, however, can encourage background awareness of situations which could trigger the need for person-to-person communication. Some projects have also allowed users to leave short audio artifacts that become part of the shared audio for the immersive media environment [18]. We could imagine this to be useful in some scenarios where micro-coordination would be used as well.

- The notion that presentation of information can be organized by context is relevant to a consideration of supporting fluid organization. This is challenging, however, since context in the real world is both complex and ambiguous.

- A challenge for researchers of immersive systems is related to the evaluation of design - questions of how to reliably create these audio environments and evaluate their use are a common thread in discussions of these systems. In most systems, researchers have struggled with the question of how to determine what kinds of sound cues are most effective for various design goals.

- An interesting architectural choice for immersive systems is the decision on whether the audio experience is created on the server, or through a user-carried device. Audio Aura for example, uses a central server to generate the experience, and sound is delivered wirelessly to headphones. A Walk in the Wired woods uses a sensor-enabled mobile to generate the audio. The same design choice can carry over for audio information access on the mobile.

- A design approach that differentiates the audio scanning systems described in this thesis was that Audio Aura avoided the question of reliability in playback by assuming in the design that the cost of missing information was negligible. The expectation was that the system would not be used to deliver information in which receipt was critical. Audio scanning in the Hearplay system will be based on the idea that explicit user interaction will be required when receipt is critical.

## 2.3 Audio interfaces to support micro-coordination

In this section, I focus on systems that allow synchronous audio interaction. By creating an always-on messaging environment, synchronous messaging could be particularly relevant for micro-coordination in mobile usages. Several systems have expanded the paradigm past the person-to-person commercial push-to-talk applications, however. The Thunderwire system is perhaps the most cited synchronous communication system, that connected a small group scattered between two buildings. I contrast a project created within the MIT Media Lab that address the design problems of synchronous communication on the mobile, Simphony.

### 2.3.1 Thunderwire

Thunderwire, an audio-only communication system similar in concept to a conference call or a shared telephone party line, was created to enable communication between a small group spread throughout two buildings [19]. The system was created to join all the audio heard in the group's personal environment together, not strictly to enable one-to-one or one-to-many personal communications. Audio captured by the system was simultaneously heard by all group members, allowing an always-on connection and shared audio experience. A field study of a deployment in a group of 10 was conducted over the period of two month to investigate the systems potential for computer supported collaborative work.

Thunderwire had several important system characteristics:

- As mentioned above, the audio recording/playback function for Thunderwire was set in fixed locations.

- Thunderwire used high-quality audio, and all sounds captured in the space could be

heard - whispers, background noise, and so on. The system had an indicator light to show that the system was active, but had no visual interface.

- All messages were public.
- Lurking was possible in this system. Users used desktop microphones, headphones, and controllers with three settings: off, listen-only, and on.
- Audio was not recorded in the system. Therefore, previously heard audio could not be reviewed.
- People could connect or disconnect anytime they wished. This was indicated by a barely audible click, and there was no way of knowing who was listening without asking.

### 2.3.2 SimPhony

Simphony is a mobile voice communication system that was implemented on iPaqs connected over 802.11b wifi. The system was designed to be deployed for a group, and deployment was studied in a microchip fab lab. The user interface metaphor was similar in concept to a 'voice instant messaging' client. Users could identify one or several people and send a voice message asynchronously. If a preset number of messages were sent and received in a certain amount of time, the system would transition automatically to synchronous full-duplex conversation. The system also allowed a user to preview an incoming conversation to decide if to transition to an incoming conversation [20].

To summarize, Simphony's important system characteristics are as follows:

- SimPhony was fully mobile, and could be used anywhere where WiFi was deployed.
- SimPhony allowed both person-to-person and multiperson conversation.
- Using a visual interface, it was possible to see who was participating in a conversation.
- The system automatically transitioned from an asynchronous voice messaging to a synchronous voice chat.

- Incoming messages could be previewed to allow users to exit synchronous chat.
- Audio cues accompany users connecting, disconnecting, and receipt of message.
- Speech recognition is used to control the interface, allowing the user to connect, disconnect, listen to and reply to messages.

### 2.3.3 Other systems

Several other systems used audio as a secondary channel for group awareness in conjunction with synchronous text or visual communication mechanisms.

ChatAmp was a system that attempts to use music as a channel for awareness of activity in a group text based chat [21]. Each user was mapped to a single instrument in a song. As a user entered a message, the music associated with that instrument would start, and slowly fade with inactivity. The position of messages in chat window also followed the music of each instrument. Study showed that music helped users quickly judge the amount of activity in a chat space, but it proved difficult to learn the association of music to individual users. Silence was perceived as awkward, creating the pressure for users to enter messages to break the silence.

Talking in Circles was also a mixture between a visual and audio interface, that allowed users to capture the experience of moving between conversations that were separated in space. As a participant, represented by a circle, approached a group of circles the audio conversation within that group became increasingly audible. The visual interface allowed users to see both who was active in the system and the social groups that were formed in conversation [22].

#### Lessons learned from synchronous communication systems

- Unexpected usage patterns emerged from the study of Thunderwire. Some coworkers, for example, wanted to share music through the system, but found this would drown out conversation. The group frequently used the system to coordinate physical interaction, often inquiring about co-workers locations. Looking back, we can see that these appropriations anticipate the need for social sharing of audio media as well

lightweight coordination mechanisms. These identified needs were part of the design goals of Hearplay, the first implementation of audio scanning described in chapter 4 of this thesis.

- Media annotation, file transfer were not included in SimPhony, but these were also noted as possible needs for workgroup support.

- Several systems have used audio as a secondary channel for group awareness. One such system (ChatAmp) used text as the primary communication channel, another system (Talking in Circles) used voice as the primary communication channel. On a mobile, however, given the small amount of screen real estate, such spatial visualizations would need to be redesigned for the smaller screen.

## 2.4 Audio to support fluid organization

Systems that provide concurrent access to audio information services are relevant to support for fluid organization. By providing ubiquitous audio access to information sources such as calendar information, messaging, and task management, mobile workers can organize and access information outside of the office. This section will focus on two relevant projects in this area, Nomadic Radio and Impromptu.

### 2.4.1 Nomadic Radio

Nomadic Radio was created to unify access to remote services including email, news broadcasts, calendar and voicemail information on an audio-only wearable device [23]. The system relied on voice recognition to access individual categories of information and control the interface. Attributes of incoming messages, such as time or arrival, were communicated using spatial audio. Speech synthesis was used to deliver text messages, such as incoming email. A dynamic alerting strategy was used to create increasingly obtrusive alerts based on dynamically identified priority of incoming messages. Therefore, the same message could be repeated at a later time, depending on its priority.

To summarize the system characteristics:

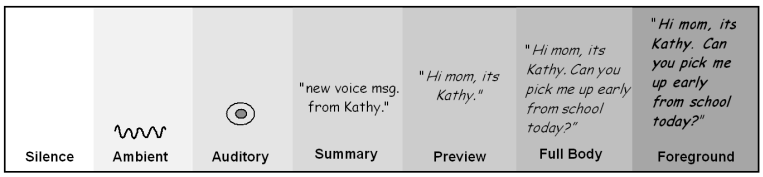


Figure 2-1: Scaling of an important incoming voice message at varying levels

- The system operates as a wearable device connected to a speech recognition component and a content aggregator.
- The system is audio-only.
- Nomadic Radio was an individual audio appliance.
- Nomadic Radio does not address the problem of synchronous communication, but serves only as a wearable audio information client.
- Speech recognition is used to control the interface, allowing the user to connect, disconnect, listen to and reply to messages.
- The dynamic alerting strategy is built around a notification strategy that uses a series of alert until a message is processed.

### 2.4.2 Impromptu

Impromptu is an IP-based mobile audio application service that is implemented on a Linux-based iPaq with 802.11b connection [24]. The mobile device allows support of multiple services, including radio, news (speech synthesis from text), music playback, telephony, asynchronous chat, and baby monitor. The user interface metaphor includes interaction techniques of conventional windowing systems, including opening an application, minimizing to background, and closing an application. There are several important system characteristics:

- The system is audio only.
- Only one application is active at a time.

- However, some applications can run asynchronously in the background and trigger an alert.

- The system includes both asynchronous and synchronous audio communication.

### 2.4.3 Lessons from audio information services

- With voice interaction, a physical button is an important mechanism for feedback. Nomadic Radio had fewer buttons for use than the imPromptu system.

- A possible criticism of the Nomadic Radio design was that though the user interface metaphor was novel, it was perhaps too abstract (similar to an immersive audio media space) for usability. The Impromptu system attempted to use a metaphor that was closer to familiar desktop user interfaces, transferring the Windowing system to the mobile.

- Alerting was a common needed characteristic for both of these systems. It was not enough simply to use voice recognition to switch between applications. Alerting was necessary to monitor changes in the virtual environment.

## 2.5 Summary of related work

Several high level observations can be made to suggest how immersive systems, synchronous communication systems, and audio information systems intersect.

- The choice of the user interface metaphor is important for creating a cohesive and coherent user experience. Most systems that were identified attempted to balance the familiarity of a user interface metaphor with flexibility and the need to minimize intrusiveness. The immersive systems and Nomadic Radio were ambient in nature, whereas Impromptu took the user interface metaphor of desktop windows, SimPhony adopted the instant messaging interface metaphor,

- In the majority of these systems, the primary benefit to the user was enabling person-to-person or group communication. Awareness of background events and access to

information services can be considered secondary in importance, since even the systems that focused primarily on this functionality identified the need for synchronous interaction in future work or were designed to support social interaction in other ways.

- These systems have shown that audio-only interfaces are technically viable. Nevertheless, a graphical user interface can help provide contextual information - specifically related to audience and context - that is difficult to do in an audio-only application. (As an aside, one could wonder if this is not a factor in the slow commercialization of such systems.)

- Several systems have identified the emergent need for sharing of media content - including audio files, music, and other information - on what could be considered applications oriented towards voice communication.

- A recurrent architectural consideration is whether audio services are created and managed from the server, or whether they should be provisioned on the client side as well. This of course does not apply as much for the systems that only provide synchronous audio communication.

# Chapter 3

# Audio Scanning

Just as mobility can produce new influences on social interaction, new user interface metaphors can optimize these new forms of interactions. To do this effectively, application designers must balance the acceptance of pre-existing user interfaces while finding new ways to support sporadic, lightweight interaction. The design challenge is to make mobile audio interfaces more useful in the real world while recognizing that tools for communication must also adapt to changing interaction demands.

One strategy is to appropriate an existing audio information interface metaphor and extend this to mobile usage. In this chapter, we pursue this strategy by describing the entailments of the radio scanner user interface, and identify how this interface could be adapted for mobile usage. Audio scanning appropriates the interface from an audio-only information appliance to help listeners maintain sporadic attention from multiple synchronous audio sources.

## 3.1 Radio scanners

Emergency band radio scanners have been used by first-responders as well as civilians to monitor non-broadcast local radio communications. Non-broadcast radio systems do not transmit a signal continuously. As a matter of protocol, to keep a communications channel open, messages in emergency response channels tend to be short. The FCC has divided the radio spectrum so that high-priority communications are protected from interference by

other broadcasts. The division of the radio spectrum into channels creates something of an expectation of the type of message that will be communicated in any given band.

Figure 3-1: A 1970's era radio scanner uses simple toggle switch for channel inclusion in scan.

If the user listens to only a single channel, time will pass between transmissions while other channels may be active. This allows the scanner to briefly tune to a given channel and detect an audio transmission above a cutoff strength. If no signal is present, the radio scanner will proceed in a given order to the next channel.

The interface for the radio scanner, particularly the first keyboard-programmable models that appeared in the 1980s, required minimal interaction. The listener would select from up to 8 to 20 channels for listening, depending on the constraints of the device, using a single touch key to toggle select which channels were scanned.

## 3.2 User experience

Audio delivery is bursty in each channel, and there may be long periods of silence if only listening to one channel. The burstiness allows a scan to create a stream from various sources, skipping only to those channels where audio is currently broadcast. Burstiness also serves as an important cue that the channel selection should be narrowed or widened to

change the rate of flow of information delivery as desired.

By listening to the sporadic playback over a period of time, a listener can learn to gauge the amount of local activity and easily distinguish the occurrence of an emergency event from a quiet night, even if first responders were scrambling the signal or using a private code.

In our terms, this simple interface allows for micro-coordination of activities (for first responders), with support for continuous partial attention (for all listeners) for a range of synchronous audio channels. The radio scanner can be used in automobiles, in the office of first responders, and even in homes, suggesting that this is a suitable model for alerting in other domains including fluid organization of workflow. The use of the radio scanner is an encouraging indication that audio can support these goals in other information appliances including the mobile phone.

To test this hypothesis, this thesis extends this audio user interface metaphor, which I call "audio scanning". The appropriation of the interface is not direct, however, since the capabilities, limitations, and usages of the mobile device are different than those of the non-broadcast radio scanner. Table 3.1 identifies a range of possible features applying to data types, user interaction, and system operation of the radio scanner that can be used for interface adaptation.

## 3.3 User interface metaphors

Though the idea of an 'audio user interface' perhaps needs no lengthy explanation, it is perhaps useful to discuss in more detail what is implied by the idea of an 'audio user interface metaphor'.

The cognitive idea of 'metaphor' itself can be complex and is not accepted without controversy in some quarters; what is useful for our purposes is the idea of metaphor as a mapping function that translates some aspect of a source domain (typically an embodied sensation, movement, or the structure of some physical or real world arrangement) to the target domain. [25, 26] Metaphors themselves can be quite simple or highly complex. The use of metaphors is largely unconscious, allowing a large number of conclusions and

Table 3.1: User Interface Features of the Radio Scanner

| **User Interface Features** | **Applies to** |
|---|---|
| All communication is spoken audio. | Data type. |
| Audio communications are short by convention. | Data type. |
| Audio communication is synchronous and not recorded. | Data type, Operation. |
| A channel is defined by the point of origination. | Data type |
| All communication has exactly one channel. | Data type |
| A listener can select a channel to monitor by pushing a button. | Interaction |
| A squelch knob allows the user to filter out audio below a given strength. | Interaction |
| The user can set how quickly to finish the sweep. | Interaction |
| The radio monitor tunes to each selected channel frequency in turn, and plays audio if it is broadcast above the squelch level. | Operation |
| After the radio monitor completes a sweep, it returns to the beginning channel. | Operation |

relationships to be the basis of reasoning. The expression `life is a journey" for example, allows us to bring to mind many possible ways in which to understand the experience of life, in a very encapsulated form.

Whether or not metaphor has fundamental importance as a basis for cognition and reasoning, as has been argued by Lakoff and others, the concept of a user-interface metaphor has proven useful for both the study of human-computer interaction and for application designers attempting to create intuitive and meaningful user interfaces.[26] One useful aspect of metaphors is that they can allow users to learn means of interaction with the computer much quicker, since the use of metaphor creates expectations on how the target domain works, even if the user has no prior experience with the target domain. For data storage on the computer, for example, the filing metaphor is often used. By recognizing the metaphor "A Folder is a Container", the user knows that data in the data storage system for example can be nested and will expect that some items in the storage system will contain others. To the extent the target domain behaves according to the expectations suggested by the metaphor, the user interface will be experienced as intuitive.

### 3.3.1 Audio user interface metaphors

Examples of graphical user interface metaphors are common. In fact, most users require these metaphors to use a computer. Audio user interface metaphors exist as well. There is a distinction that can be made between the use of audio *in* user interface metaphors, and audio *for* user interface metaphors. Part of the reason that the audio experience on mobile devices can be so unsettling is that these can conflict, leading to confusion.

Audio can be used in an existing metaphor to fulfill user expectation and strengthen the use of an underlying metaphor. It is common in some mail programs, for example, when sending an email to hear a "whoosh" as the email is sent. The grounding metaphor at work here can be labelled "Sending a message is throwing an object". The use of audio in this way re-enforces user expectation.

Audio *for* user interface metaphors builds upon these grounding metaphors in more complex ways. There are at least three metaphors that I can think of how prior work has used audio user interface metaphors in this way.

- One metaphor can simply be stated "The computer is a world". The expectation that this creates is that there are objects that interact that will have audio manifestations. This user interface metaphor certainly underlies the *Audio Aura* project and *A Walk in the Woods*, and is therefore probably useful for supporting continuous partial awareness in general.

- Another commonly used audio interface metaphor is simply "The computer is a person that understands spoken language". This creates a very different set of expectations on how audio can be used. Projects that have used speech recognition for command and control that were identified in the previous chapter include *Nomadic Radio*, *SimPhony*, and *Impromptu*. I believe that these are appropriate in cases when two-way sporadic interaction with the computer is required. Speech recognition has progressed to the point where such application is useful, but researchers have long accepted that the problem of speech understanding does not reduce simply to the problem of recognizing spoken instruction [15].

- The final metaphor is "The computer is a telephone (or radio, etc)". Clearly, this interface metaphor creates expectations for the instrumental use of the computer for communication. *SimPhony* or *Nomadic Radio* use this metaphor as well. These projects are interesting because it illustrates that organizing metaphors can overlap, creating both new potential usages and perhaps complications as well.

The use of almost any user interface metaphor will produce complications. Used as an organizing metaphor in audio, when compared to the same metaphor in organizing graphical user interface metaphors, the metaphor "The computer is a world" seems more problematic, for example The source of a sound can be more ambiguous when we want to identify an object that suddenly appears, when compared to scanning a visual scene. It is also hard to estimate how many sources of audio there are in the audio interface as compared to objects in the visual interface. The metaphor of the computer as a person capable of spoken language, similarly, has produced complications that are quite extensive and widely known.

### 3.3.2 Creating novel user interface metaphors

These two audio interface metaphors are certainly not exhaustive, and can be quite specialized for specific usages. Audio Hallway explores the metaphor that "Listening to news is walking down a hallway" [27]. However, a reasonable design heuristic is that being able to make a an explicit statement of the underlying user interface metaphor can be a good strategy for ensuring that the application of the user interface is both coherent and useful.

The audio user interface metaphor that is being explored in this thesis can perhaps be stated "The mobile is a noisy radio". The user interface metaphor is purely instrumental; it does not attempt to map the audio interface to the virtual world or intepret spoken language. My hope is that metaphor can be both encompassing and useful for supporting mobile interaction. A radio scanner is even better as an interface metaphor for the practical reason that there are certain expectations created by a broadcast radio metaphor that would be hard to arrange on the mobile device (the idea of DJs, of a broad range of channels, of program formats, and so on). Instead of relying on audio cues to indicate activity in the real world, I believe that increase or decrease of activity in the radio scanner can convey this awareness.

### 3.3.3 Identifying entailments

To understand how the radio scanner user interface is applied in a specific application, it is necessary to first identify a range of elements that can be understood easily by the user in the new application. An entailment is simply a description of a particular thing some signifier in a user interface (such as an interface element or user action) implies about the signified [25]. As an example, in a desktop system with its use of a filing metaphor for data storage, the following are noted as applicable entailments:

- There are files in the data storage system.
- There are folders in the data storage system.
- Files can be placed in folders.

Some entailments are possible but not applicable

- There are filing cabinets in the data storage system.
- There are drawers in filing cabinets.
- Folders can be placed in drawers.

With the creation of a novel user interface metaphor a heuristic that has been suggested is to ensure that all entailments are identified before application [25].

Here are possible entailments that can be identified in the audio scanning metaphor for content distribution and presentation. These entailments are drawn from the preceding list of user interface features in the preceding table.

- There are channels in the distribution system.
- A single button refers to a single channel in the distribution system.
- A channel can be selected with a single button press in the distribution system.
- Content can only be in one channel at a time in the distribution system.
- There are channel sweeps in the distribution system.
- The regularity of channel sweeps can be chosen in the distribution system.
- There is a squelch knob in the content distribution system.

In the following chapters,the audio scanning interface will be applied in two very different use cases, each of which needs to review of audio information on the mobile. The notion of entailments will be applied to show what parts of the metaphor are applicable to each use case.

# Chapter 4

# Hearplay

This chapter describes how audio scanning can be applied to aggregate and share mobile audio annotations within a group, through a system called Hearplay. This chapter is divided into four sections:

- The first section describes the design motivations in detail.
- The second section outlines the specific manner in which audio scanner was applied.
- The third section describes the technical implementation details of the Hearplay system.
- The final section compares the system to previous work and offers a design critique of the system.

## 4.1 Motivations

Mobile devices can be noisy, distracting, and intrusive. What is more, audio alerts on the mobile can also be incoherent and confusing. Part of the reason for this is that the design process rarely spans multiple audio use cases on the mobile, focusing too narrowly on how audio can be used for one functional purpose without consideration of how multiple applications are used in tandem while mobile. When application designers support aspects of the combined audio user experience, they seem to focus on either individualizing the audio user

experience to distinguish it from other devices, or to silence the device completely when audio interruption could be disruptive. These issues are only part of the problem in improving the user experience. How can audio in all its presentation forms be better aggregated and adapted for mobile users? How does the selection of a user interface metaphor affect the user experience of mobile audio? Audio scanning is one design approach to this problem that is well-suited for mobility.

The expected usages of audio on a mobile device relate primarily to audio communication (person to person messaging), audio alerting for notification purposes, and playback of stored audio content on a mobile device. Less frequently, recorded voice or audio is used as an input method (perhaps most frequently, speech recognition for voice dialing or system control). Each of these audio usages in the interface have characteristics which are worthy of individual consideration before considering the application of audio scanning to the problem of aggregating audio presentation on a mobile device.

### 4.1.1 Need for managing alerts

Audio alerts are commonly used in mobile applications. The purpose of an audio alert is to draw the user's attention for an operative purpose at an opportune moment. On the mobile, an sound can alert the user to a phone call, incoming mail, or draw the user's attention when a background process in an application is completed. This model is effective and easily understood when only one application is being used at a time. But given the increase in the number of mobile applications available to a user, a mobile user is likely to use several applications at once. If an alert arrives, the user can be confused by which application produces the alert or for what reason. The distraction is compounded when an alert arrives during a phone call, or any other time when the user does not wish to be interrupted.

Sounds can also be inconsistent from one application to another. Audio alerts may therefore increasingly overlap both in form and timing, create ambiguity, and if designed poorly serve more as a possible distraction and source of confusion.

### 4.1.2 Need for lightweight communications

Mobile devices are also used for person-to-person and for group communication. The most common form of person-to-person communication, a simple phone call, is time-consuming and requires full attention from the person taking the call. Push-to-talk is commercially available and widely accepted, especially among on-the-job support workers who must be in frequent touch. But in public contexts push-to-talk can be annoying to others, and when the user is trying to focus, voice can also prove to be as interrupting as an unwanted alert.

### 4.1.3 Need for supporting audio input as a datatype

In some respects, audio is an easier input method on the mobile. Consider the difficulty of text entry and of moving data from one application to another on the mobile device as compared to the desktop. On the desktop, we might receive a URL in an RSS reader, open this in a web browser, cut text from the web page and paste into an open IM conversation, and finally cut the entire conversation and paste it in an email message, add a short note and send it to someone around the world.

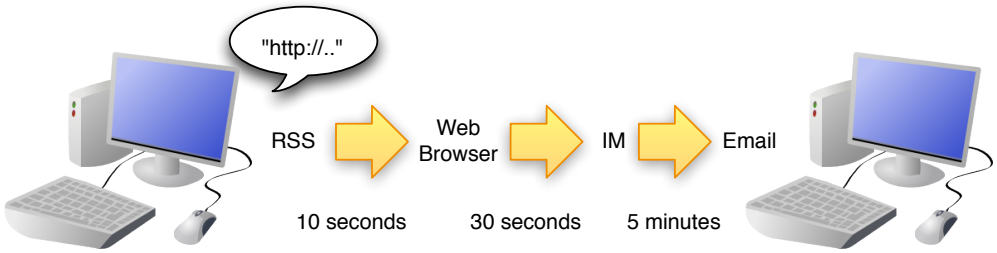


Figure 4-1: Text information can be transferred effortlessly and quickly between desktop applications

On a mobile, this complicated scenario is too unwieldy to consider using cut-and-paste interaction on the small screem. Audio might prove to be a more lightweight control method on these devices. Speech recognition is increasingly used on mobile devices as processing power and digital storage improves. We could imagine speech recognition to be used to achieve the same actions on the mobile as we might use the mouse on a desktop, in a command and control method.

However, though speech recognition has utility, simple audio annotation is an area that is largely unexplored. Though the mobile can be used as an audio recorder, audio information is still poorly integrated as a data type with other information in the mobile platform. It is difficult for example, to associate an audio note with other information, such as a contact listing or web page. If the purpose of audio is simply to remind ourselves of some bit of information, the extra verification required for speech recognition is unnecessary and possibly taxing, especially in less than ideal acoustic circumstancees.

## 4.2 Adaptation of audio scanning for mobile devices

In the previous chapter, the potential of the radio scanner was introduced as a user interface metaphor to support mobile audio devices. The goal of audio scanning is to support mobile interaction patterns in a consistent user interface while allowing aggregation of a variety of audio sources.

As pointed out in the previous chapter, identification of the entailments of the user interface metaphor is a useful early goal when applying a new user interface metaphor to a target domain. To avoid confusion for the new user, most of the entailments of the interface metaphor should be mapped to elements in the target domain.

To give a vivid illustration of this point, imagine a radically different metaphor for data management - for example, the metaphor of data management system as a kitchen cutlery drawer. We might map the entailment of stacked and sorted cutlery in a drawer divider to imply the necessity of a homogenous, sorted set of organized data. This would be a very different entailment than the heterogenous set implied by the file folder entailment in the filing user interface metaphor. For any novel interface metaphor, the designer would want to be very clear on how the entailments of the new metaphor would map to the data types, interaction goals, and underlying processes of data management.

Not all of the entailments of a user interface metaphor need be applicable, but identifying the mapping of the metaphor to the domain early on can help evaluation in several ways:

- It forces the designer to become aware of the utility and limitations of each entailment of the user interface metaphor, avoiding reliance on ambiguous qualitative reactions

Table 4.1: Applying entailments of Radio Scanner interface metaphor in Hearplay

| **User Interface Metaphor Entailment** | **Applies to** |
|---|---|
| A radio scanner plays audio in 'bursts'. | Applied in Hearplay. The 'burst' is the fundamental message in Hearplay. |
| A radio scanner plays audio in 'channels'. | Applied in Hearplay. Each channel is mapped to a type of content. |
| A single button turns a single channel on or off in a radio scanner. | Not applied in Hearplay. The device button control is mapped to navigation within a channel in Hearplay. |
| Content can only be in one channel at a time in the distribution system. | Applied but not enforced in Hearplay |
| There are channel sweeps in the distribution system. | Applied in Hearplay |
| The regularity of channel sweeps can be chosen in the distribution system. | Not applied in Hearplay. The regularity of sweeps is determined from time of receipt of new material |
| There is a squelch knob in the content distribution system. | Not applied in Hearplay. |

to the experience as a whole.

- It can assist in evaluation by narrowing the scope and allowing each entailment to be evaluated in turn.

- It can expose some blind spots early on of a user interface in terms of usability.

The following table summarizes the entailments of audio scanning and their application in Hearplay. The remainder of this section will discuss in several entailments in greater detail to more fully describe the desired user experience.

There are three fundamental entailments that translate directly into entities in Hearplay: bursts, channels, and sweeps. The relationship between these entailments shows how audio scanning supports several aspects of mobility. The following discussion of each entailment is followed by a diagram that shows the audio scanning data model in greater detail.

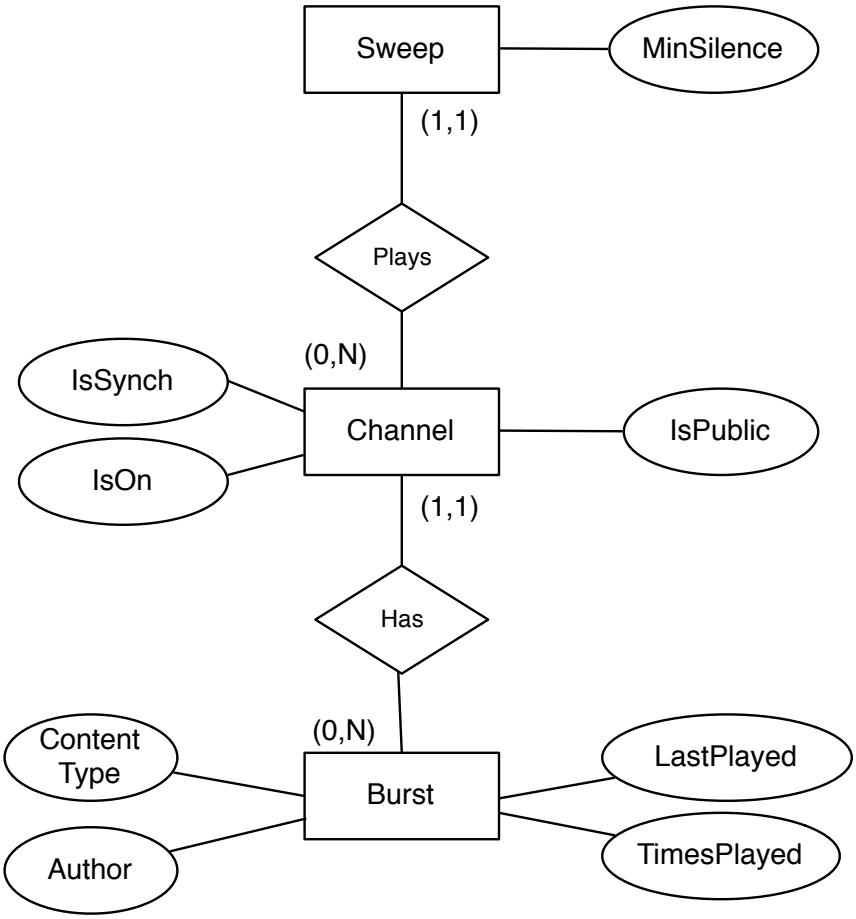


Figure 4-2: Audio scanning entity relationships in Hearplay

### 4.2.1 Bursts

A radio scanner plays audio in bursts. Application of audio scanning as a user interface metaphor in Hearplay used the *burst* as a fundamental content type in the user interface.

- A burst must have a length (how long the scanner needs to play the burst). A burst is expected to be short. However, this is not currently enforced by either the radio scanner or Hearplay.

- A burst is assumed to be audio. It can also be text or an image with audio associated to it. Though video was not supported in Hearplay, since it has a length, it could be used as a burst if supported. Text or images alone however would need a display length to be associated before being usable in this interface. In hearplay, text messages were delivered in audio form using text to speech.

- A burst is associated with a source (the 'author' or originator).

- Other metadata attributes can be associated with a burst: a timestamp, a semantic label, a description, and so on. The interface itself does not prescribe what metadata must be associated with a burst, but this can be required by a channel. In Hearplay, there are 5 burst types which are associated how the burst was created: email, calendar, web log (an audio annotation of a web page), music, and voice notes (an audio recording not associated with any content).

- A burst can be joined with other bursts. When annotating audio in Hearplay, the audio annotation is appended to the original audio.

### 4.2.2 Channels

A radio scanner plays audio bursts in channels. A radio scanner will play only one channel at a time and only one burst at a time. Hearplay operates in the same manner.

Channels have attributes that are not associated with bursts. For a radio scanner, the only attribute that a channel has by definition is a defined radio frequency range. The use of that attribute, including who may speak on the channel or for how long, is fixed by legal or user convention. The FCC might mandate that only certain groups can use a given

channel, or users can simply decide to move to a certain channel to conduct and monitor a conversation.

In general, the attributes of a channel allow bursts to be organized in the audio interface. In Hearplay, each channel is associated with one of the 5 burst types. Bursts can be added to a channel automatically by other programs (for email and calendar) or manually by the user (for web-log, music, voice notes). We can imagine it to be possible that a channel could be created by a combination of metadata that does not refer to a single burst metadata attribute, especially if the amount of bursts in Hearplay is large.

A channel could place other restrictions on bursts as well, although this was not reflected in the Hearplay interface. A channel could set a maximum length for bursts, for example. This would prevent single burst from monopolizing channel playback. These kinds of restrictions would also be channel attributes.

For Hearplay, attributes may refer as well to the notion of permissions to listen or to broadcast bursts on the channel. Email and calendar are implemented as private channels (no one else but the intended user can listen or broadcast on these channels). Hearplay shared web log, music, and voice notes are implemented as public channels. Anyone who is listening to Hearplay can listen or broadcast on these channels.

Other attributes alter how and when bursts are presented. The channel attribute of this kind that was implemented in Hearplay relates to the division between asynchronous and synchronous channels. An asynchronous channel simply means that bursts in these channels are captured and that the mobile will play them at some point after it receives them. The strategy for playback of asynchronous channels is discussed in the section on *Asynchronous presentation strategy* below. Asynchronous channels will only play bursts after an sweep has started. Each asynchronous channel will therefore act as a message queue, and store bursts to potentially play back at some later point.

Synchronous channels will play audio bursts as soon as the mobile receives them. 'Bursts' in these channels are not typically recorded (with one exception: when the scanner is playing asynchronous audio, synchronous audio is captured for playback after the 'burst' playback ends. This will be discussed in the implementation section). Synchronous channels, therefore, do not repeat content. Synchronous audio does not trigger asynchronous playback in

Hearplay, but this further variation in parameters has been implemented in the Hearwell system, described in the following chapter. Voice notes are treated as a synchronous channel in Hearplay.

### 4.2.3 Sweeps

A radio scanner plays each channel consecutively during a sweep. The role of the sweep, then, is simply to specify the order in which active channels are played. A radio scanner can be in only one sweep at a time and in only one channel at any given time during the sweep. At any given time, the radio scanner can be in one of two states: inactive, or in a sweep. Hearplay operates in the same manner.

In the use of audio scanning, Hearplay operates differently than the radio scanner in one regard: unlike Hearplay, each channel of a radio scanner is purely synchronous. If a radio scanner is playing audio in one channel, and there is a burst that arrives in another channel, the radio scanner will miss at least part of the later burst. Since the convention among users of the radio scanner is that bursts are short, the likelihood of this happening is usually small. Nevertheless, if two bursts were to arrive in two separate channels simultaneously, and the burst that is not active is the same or less in length than the first burst, it will not be heard. Hearplay avoids this situation by capturing synchronous audio that arrives during active playback. Once the active channel finishes playback of the burst, the captured burst is played. To summarize, a synchronous channel can interrupt playback in a sweep (but not a burst) or play when audio is silent.

Secondly, in Hearplay, the default is that asynchronous audio plays during a sweep. As the sweep progress, each channel in turn presents the bursts that are in the message queue and ready for playback. The details of this presentation strategy are presented in the next section.

To summarize the preceding discussion, the following flowchart shows the details of how all arriving bursts are processed in the system.

The only attribute for sweeps that was used in Hearplay is a constraint on the minimum amount of time to guarantee silence between sweeps. This attribute is an important user control when bursts arrive frequently. It also allowed the system to switch quickly from a

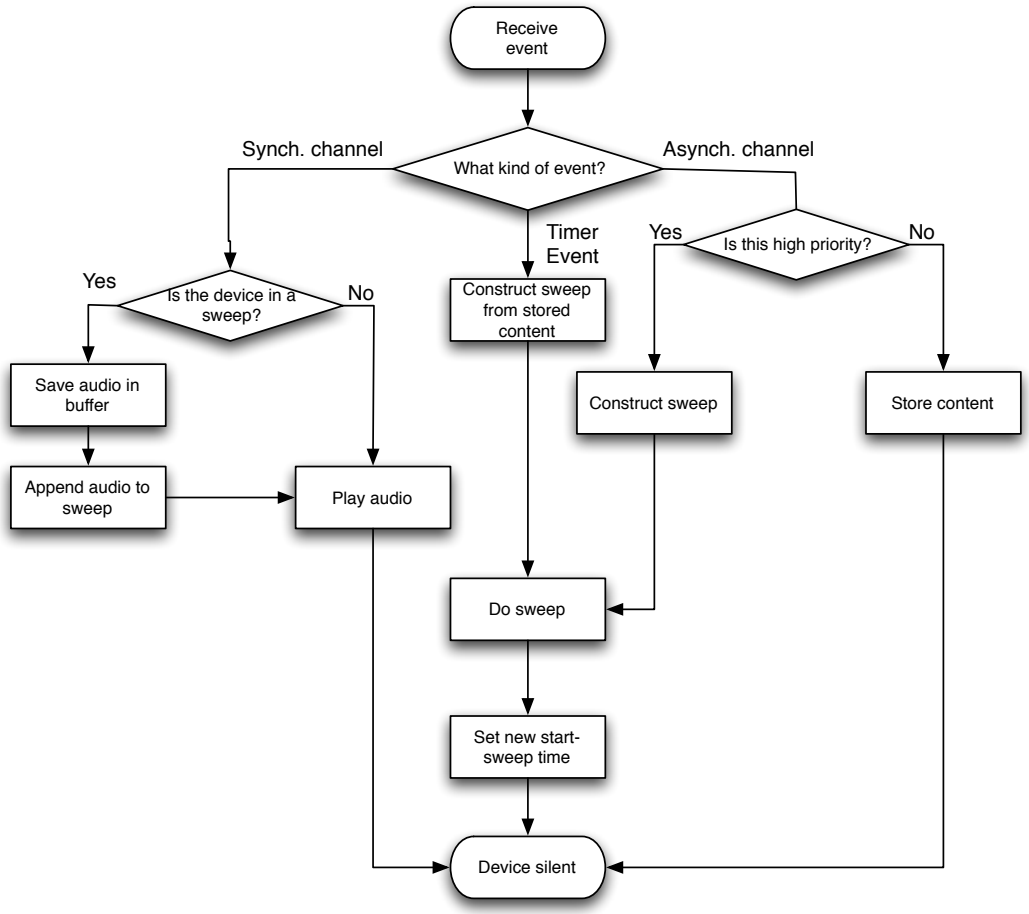


Figure 4-3: Flowchart for burst arrival

demo mode, where it is necessary to have very short times between silence, to a working mode, where longer silences are desirable. If uninterruptable silence is desired, an interface refinement could allow the user to change a synchronous channel to an asynchronous channel on the fly - in this way, no interruption would be allowed when the device is silent.

## 4.3 User experience and user interaction

The experience of listening to Hearplay is intended to be similar to the experience of listening to a radio scanner, in that the user passivley hears bursts of audio followed by periods of silence. This user interface metaphor requires no explicit user interaction to listen to incoming audio. Hearplay, as an always-on audio information appliance, does not require user interaction to play received audio from any source.

Figure 4-4 shows a graphical representation of the audio experience delivered by the Hearplay scanner. A sweep is followed by periods of silence. The scanner uses audio cues, or audio icons, to provide structured to the aggregated audio. An audio icon is used to start the sweep (in the Hearplay implementation, this was a series of rising tones), and an icon used to indicate the end of a sweep (a series of falling tones). As one channel transitions to another in the sweep, an audio icon marks the transition point. No explicit indication of the channel was added in Hearplay, but could prove necessary if the number of channels in the system were to increase. Figure 4-4 shows the scanner in operation, after a burst is explicitly deleted by the user, and after the receipt of a new burst. The organization within a channel is discussed below.

User interaction with Hearplay is different than interaction with the sort of radio scanner shown in Figure 3-1. In a radio scanner, the user can select which channels to monitor using a row of buttons, and use the squelch knob to silence transmissions below a certain strength. This is a useful interface for a synchronous-only audio appliance, in which the primary user interface requirements are to allow the scanner to be rapidly reconfigured for changing circumstances of synchronous listening. User configuration in a synchronous audio information appliance is limited to configuration of playback.

Hearplay creates the notion of asynchronous channels to minimize user interaction and

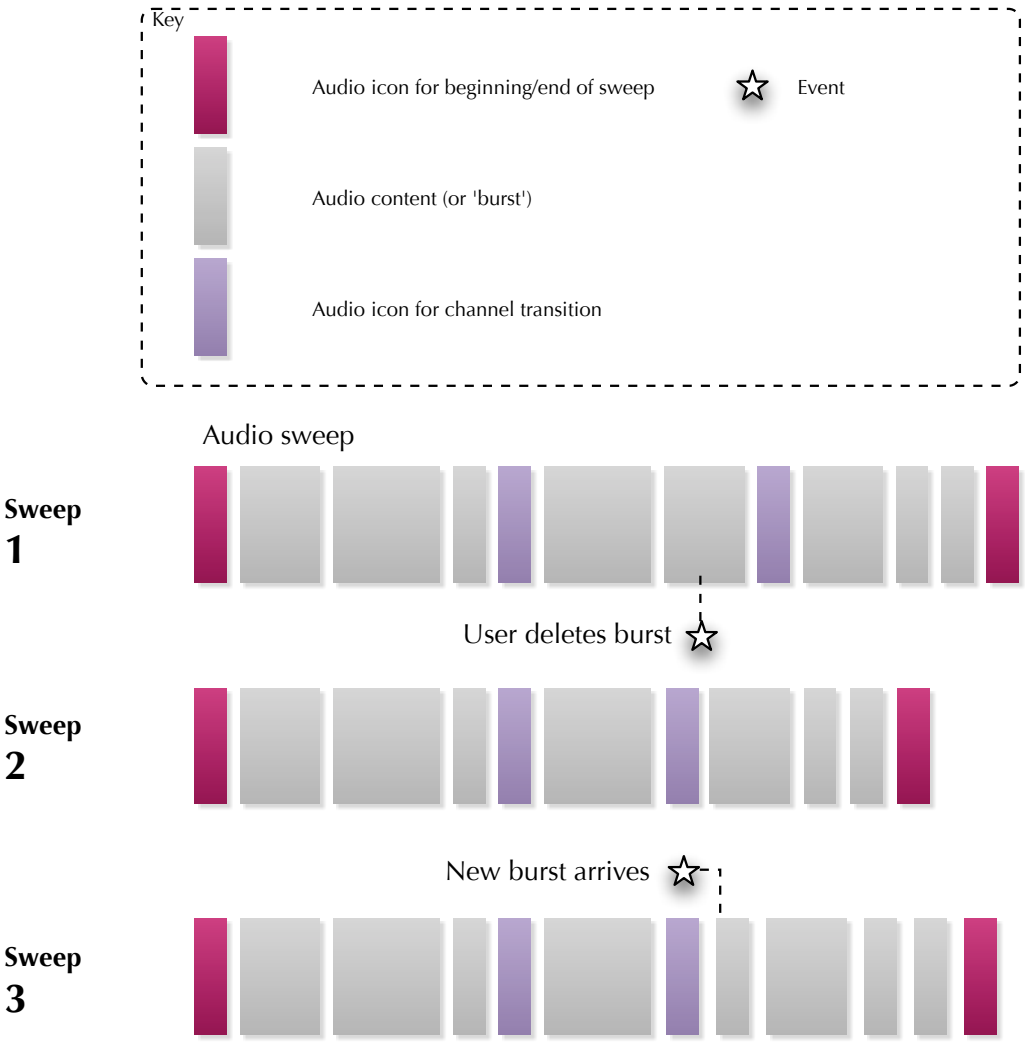


Figure 4-4: User experience of a sweep

to allow periodic information delivery. However, just as synchronous channels do not require explicit user control, playback of asynchronous channels can be passive as well. There are two impacts this creates on user interaction.

- First, a user needs greater navigational control in an asynchronous channel; the user interface must support the ability to navigate between bursts, navigate between channels, start a sweep when the device is silent, or end a sweep when interruption is not wanted.

- Secondly, a strategy must be in place to ensure that bursts in asynchronous channels are heard by the listener. The interface therefore must support a playback strategy that makes it likely (if not absolutely certain) that the user will hear an asynchronous burst. The system should also support an optional means of user confirmation to remove a burst from playback.

### 4.3.1 User navigation

Unlike the radio scanner, the user does have the ability to navigate content within the audio scanner. User interaction is possible either during a sweep or during silence. On the iPaq, the user can interact with the scanner using the 5 hardware keys, and the 5-way directional button provided, shown in Figure 4-5. Button controls are described in Table 4.2.

### 4.3.2 Asynchronous presentation strategy

In asynchronous channels, Hearplay can receive a burst without playing it back immediately. Since a user does not have to explicitly start or stop playback, it is usually not certain in Hearplay that the user has heard playback of a given burst. To compensate, the heuristic that is used in Hearplay to ensure that audio bursts will be heard by the user is to reschedule the burst after playback to replay in another sweep (unless the burst is explicitly deleted).

This leads to the question: how does the asynchronous channel decide which stored bursts to play in any given sweep? A number of possible heuristics were considered before selecting the strategy that is implemented in Hearplay.


Figure 4-5: Buttons at the bottom of the iPaq device

Table 4.2: User interface actions in Hearplay

| **Button pressed** | **During scanner state** | **Used to** |
|---|---|---|
| Record | Silence | Record and send synchronous audio. |
| Record | Sweep | Record audio annotation for confirmation. |
| Record | Outside scanner usage | Capture image and record audio annotation. |
| Toggle Left | Sweep | Navigate to previous burst. |
| Toggle Right | Silence | Start a sweep. |
| Toggle Right | Sweep | Navigate to next burst. |
| X | Sweep | Delete a burst. |
| X | Confirm annotation | Delete annotation. |
| Quit | Silence | Begin sweep. |
| Quit | Sweep | End sweep. |
| Quit | Confirm annotation | Delete annotation. Go back to previous state. |
| Arrow | Confirm annotation | Send annotation to server. |

- Strategy 1: Play all bursts in each consecutive sweep (of fixed length) until they are deleted.
- Strategy 2: Play all bursts in each consecutive sweep (of flexible length) until they are deleted.
- Strategy 3: Play bursts less frequently with each consecutive sweep until they are deleted.

To present an example of the benefits of the back-off heuristic when compared to the alternative playback characteristics, consider the following scenario:

- A sweep is 60 seconds in length.
- A burst is 10 seconds in length.
- For 10 minutes, a burst arrives every minute.
- No messages arrive after 10 minutes.
- No burst is deleted by the user.
- For strategy 3, each burst plays 1 minute less frequently with each consecutive sweep.

How would each strategy affect the operation of the system? The operation of each strategy in this scenario is described below and summarized in the following diagram.

- Strategy 1: The simplest playback heuristic would be to play all bursts in each consecutive sweep until they are removed by the user. The problem with this approach is that the length of the sweep would quickly expand as new bursts arrive, until the scanner is playing without interruption. After 6 minutes, the length of the sweep increases to the maximum of 60 seconds. If the scanner were to enforce the notion of the sweep at a fixed length, some bursts might never be played in this presentation heuristic. The channel might have to enforce a last-in-first-out (LIFO) presentation strategy, which would have one advantage: older content could fall off the sweep with no explicit user interaction. If newer content were to be explicitly removed, then the older content would reappear in the sweep. However, this strategy can easily create a user interface which is much more continuous or repetitive in playback than is desired.

- Strategy 2: This strategy behaves similarly to the first. After 6 minutes, the length of the sweep begins to increase to accomodate all incoming content. (In terms of time spent in playback, the user experience for strategy 1 and strategy 2 are identical, but the bursts presented in a sweep would differ.) In this presentation strategy, the user would be expected to have some difficulty in recognizing new bursts in the uninterrupted sweep (especially without explicit audio cues to indicate newly arriving bursts) since they would be played back less often in any given time period. We could imagine after a period of use, and no interaction, the sweep length could grow quite long indeed. With very little sweep repetition, the scanning metaphor would be largely broken; either explicit navigation would be required, or the entailment of distinct channels would have to be sacrificed entirely because channels would not release control of the sweep when playing a long stream of bursts. Users would have to break off playback explicitly. In addition, when the sweep length grows quite long, drastic disposal of the entire messaging queues or repetitive removal of content would be necessary to shorten the sweep length significantly.

- Strategy 3: In this strategy, a burst is played less and less frequently as sweeps proceed. A burst will be played less and less frequently as sweeps proceed. The timing strategy used here will shorten the number of bursts played back, producing silence after the X sweep. Nevertheless, former bursts will reappear, increasing the likelihood that the user will interact with old content, spreading out the interaction needed to clean out old content. The advantage of this strategy of this is that it allows more flexibility for both channel presentation order (allowing either LIFO or FILO presentation depending on purpose of channel) and shorter sweep sizes in general than either the first or second strategy above. Secondly, the longer silences between playback indicate something about the recency of activity, which is not conveyed in the strategies above.

Hearplay, then, implements a back-off timing heuristic to schedule burst playbacks in each sweep. To accomplish this, there are two attributes that are associated with each burst: a timestamp for when the burst was presented in a sweep, and the number of times since receipt that the audio burst has been presented in a sweep (these are shown in the entity re-

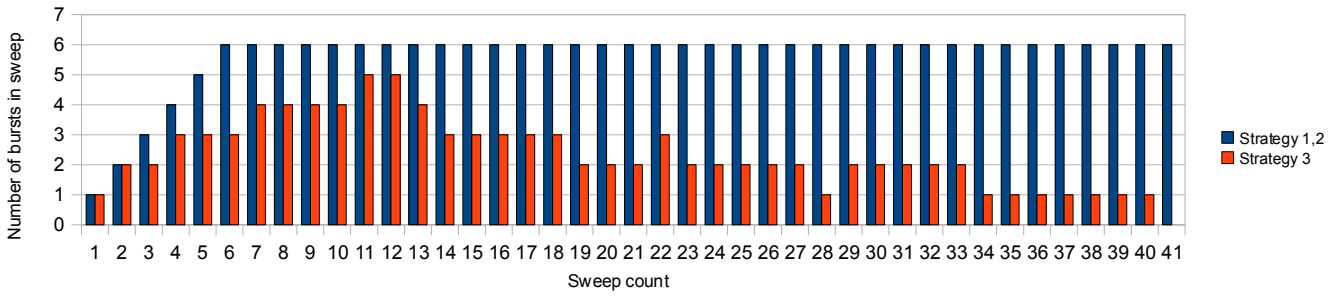


Figure 4-6: Comparison of content timing strategy alternatives

lationship diagram in Figure 4-2). The timing schedule can be determined by mathematical formula or implemented in a look-up table. In Hearplay, a backoff timer was created to calculate each consecutive interval, given the desired number of iterations until a fixed final time period. The relevant code excerpts for both the decay function and the look-up are given below.

```
void BExpDecay::SetDecay_FixedLastInterval(int iterations, int interval)
{
ASSERT(iterations > 0);
ASSERT(interval > 0);
fIterations = iterations+1;
float ratio = (float) (((float)1)/fIterations);
fCurrentIteration = 1;
fConstant = ((log(ratio))/interval);
return;
};

int BExpDecay::GetNextInterval()
{
float current = (float) (fIterations - fCurrentIteration)/fIterations;
```

Table 4.3: Back-off time interval in Hearplay demonstration mode

| **Repetition** | **Time interval (seconds)** |
|---|---|
| 1 | 1 |
| 2 | 3 |
| 3 | 5 |
| 4 | 8 |
| 5 | 12 |
| 6 | 20 |

```
int time = (int) (log(current)/fConstant);
return time;
};
```

---

The look-up table implementation would perhaps allow greater flexibility and transparency to the creator of a content channel, and allows the back-off strategy to mimic both earlier strategies if desired. For the first strategy, the timing schedule would be set constant for every playback time. The playback schedule can also be used to ensure that the burst is played only a certain number of times, and is automatically extinguished afterwards.

In Hearplay, the timing strategy is associated to the channel in the entity relationship. In Hearplay, 3 of the 4 asynchronous channels used the timing strategy backing-off from time of burst arrival. The back-off heuristic can be used in relation to a time in the future, to play certain content more frequently as the future time approaches. This was used in Hearplay in the calendar reminders channel.

For demonstration purposes of the back-off timer, it was often clearer to show how one burst would play less and less frequently until a desired maximum time, rather than showing a complex interaction of bursts in multiple channels. In demonstrations, a strategy of backing off to a maximum of 20 seconds after 6 repetitions was clear. The timing function, given these parameters, produced the time intervals of Table 4.3.

Usage of the timing heuristic showed that the schedule would have to be further adjusted to burst and channel parameters after usage data is gathered, if finer control is desired over the aggregated sweep. These parameters could include average burst length, average number of bursts received between sweeps, total time desired for sweep or channel length, and the

intent of the channel itself. Further research would be needed to automatically adjust the back-off timing heuristic to these parameters to be responsive to usage patterns and amount of stored bursts in channel queue.

## 4.4 User scenario

The following is a hypothetical scenario to illustrate how audio scanning could integrate the user experience of audio usage on the mobile in a shared group context.

*John works second shift in the IT department of a global company that provides 24 hour support. John is often away from his desk but always has his mobile with him.*

*Hearplay aggregates information that John wants to check regularly on his mobile. A few times every hour, Hearplay starts playing automatically, giving him a stream of email, voice mail, and even music. John knows that if he doesn't press a button to remove a message, it will be played at some point in the future. However, it will be played less and less frequently, so he knows that he will always hear newest information first. He often leaves the message intact to serve as a reminder, or any case when he wants to review the message later, perhaps when he is back at his desk. After using Hearplay, the short bursts of information at frequent times, the format and playback order is familiar to him. Even the experience of silence is meaningful to him - it tells him there hasn't been that much recent activity in the group as a whole.*

*Hearplay allows John to stay aware of what's going on in his group without having to constantly check email, voice mail, or stay connected on instant messaging system. John's group uses Hearplay to respond quickly to issues that might arise during the shift. One of the most common uses is as a walkie-talkie. John can simply push a button, and a short message is played to everyone online in the group. Voice messages in this channel are not recorded, so the group can use this channel for quick questions or even conversation. John can also select a person to send a private message to. If the person is away, or signed off, John knows that it will be played for them after they return. John can also attach a voice message to a web page - such as a bug report - that he browses on his mobile. The message will be shared among members of his group.*

## 4.5 Implementation overview

This section provides greater detail into the implementation of the Hearplay system. The system is composed of a server, with several programs running independently, and software clients running on a iPaq connected to 802.11 wifi. The sole responsibility of the server is to store asynchronous data for retrieval and to serve as a conduit for synchronous audio. In this way, aggregation and scheduling functionality is implemented primarily on the client.

### 4.5.1 Server

The Hearplay server operates as an communications server that coordinates messaging services between remote clients. The server stores asynchronous audio and other media files for mobile clients. Server side applications did not need modification. It was necessary, however, to have the WiFi router to be UDP enabled. As there was no multiple groups or that were not public, no specialized middleware for group management was necessary for the Hearplay demo. As the primary goal of Hearplay was to test the user interface, the task of coordination was left to individual Hearplay applications, rather than designing a server for scalability and performance.

To summarize, server side components include the following:

- A Jabber server, providing messaging capability using XMPP message transfer format.
- A Jabber client, written in Perl, on the server side acts as the conduit for messages between clients and for notification. Future work could reengineer this into a Jabber component, to allow group discovery and enforcing messaging permissions as well.
- An FTP server.
- Perl scripts that monitor for new email and changes to calendar events. These clients were derived from existing scripts running in the Speech Interfaces Group as part of the CLUES e-mail filtering system. The primary modification was the inclusion of notification to the Jabber client corresponding to the user.
- UDP-enabled wireless router.

### 4.5.2 Clients

Hearplay was implemented on Compaq iPaq (model h3650), connected to the wireless 802.11 network. There were two applications created: one was an audio annotation program that was the sole separate authoring application. The second was an application that provided the scanning interface to content created and shared within the system. Both clients were implemented in Windows Embedded Visual C++ 3.0.

#### Audio Annotation Program

The audio annotation program allowed image capture from any application running on the iPaq, with an associated audio annotation. Once this was running as a background process, the user could use any application on the mobile normally.

A sequence of three hardware buttons were used to capture information from the mobile, attach an audio annotation, and share the annotated image with the Hearplay group. These buttons are identified in Table 4.2. The user could also quit out of the process as well.

By pressing one button, the current image on the mobile display was captured. It was proven to be technically possible to capture meta-information associated with the current display, depending on the information exposed by source mobile application APIs. The only implemented meta-information of this nature was the source URL associated with a web page viewed in the iPaq web browser.

By pressing and holding the side button of the iPaq, the user could record an audio annotation to comment on the captured image. Recording would end once the side button was released. A short beep would indicate the start and stop of recording. After release, the recorded audio would be played back for user review. If the user repeated the recording process, the new annotation would replace the old. Once the recording proved satisfactory, the user pressed the send button. Both the image and the recording would be uploaded via FTP to the Hearplay host server. A control message is sent over the Jabber server Once received in full by the server, the Hearplay server would send out a notification to clients on the Jabber control channel.

### Hearplay Scanner Overview

Though the interface has no GUI control, the Hearplay Scanner program architecture follows the Model-View-Controller paradigm. The 'Model' in Hearplay is responsible for burst, channel, sweep, and other program state information. The 'View' is responsible for sweep and burst timing, burst presentation, and playing back audio in synchronous or asynchronous form. Since the system implements asynchronous, synchronous, and annotation modes, the view is responsible for coordinating these modes. The Controller is responsible for coordinating the model and view with user interaction and events from the server.

The structure of source code `#include` files in Hearplay Scanner, shown in the Figure 4-5, will convey a high-level idea of the structure of the Hearplay Scanner program. Since experimentation was required to find the best way of supporting real-time audio playback and messaging, the software architecture decoupled these interfaces to allow alternative implementations to be evaluated. As a consequence,there are several objects that are created within the global namespace. These objects are responsible for communication with the Jabber server, receiving audio, managing sweep presentation, and providing audio recording capabilities. Hearplay Scanner uses a library called `libsigcce`, a slot/signal messaging bus library, to allow these independent components to communicate.

### Model

The Model classes of Hearplay scanner are related to burst, channel, sweep, and current state information. `BMessageManager` is responsible for parsing and storing received messages related to bursts in the appropriate channel message queue.

### View

The View classes of Hearplay scanner are related to the user interface modes. The View classes also control playback of incoming synchronous audio, and recording of both synchronous audio and of audio annotations. There are three basic user interface modes: `AsynchMode`, `SynchMode`, and `AnnotationMode`. The `BMessagePresentation` class executes the burst presentation and controls presentation of a burst. The `BSweep` class or-

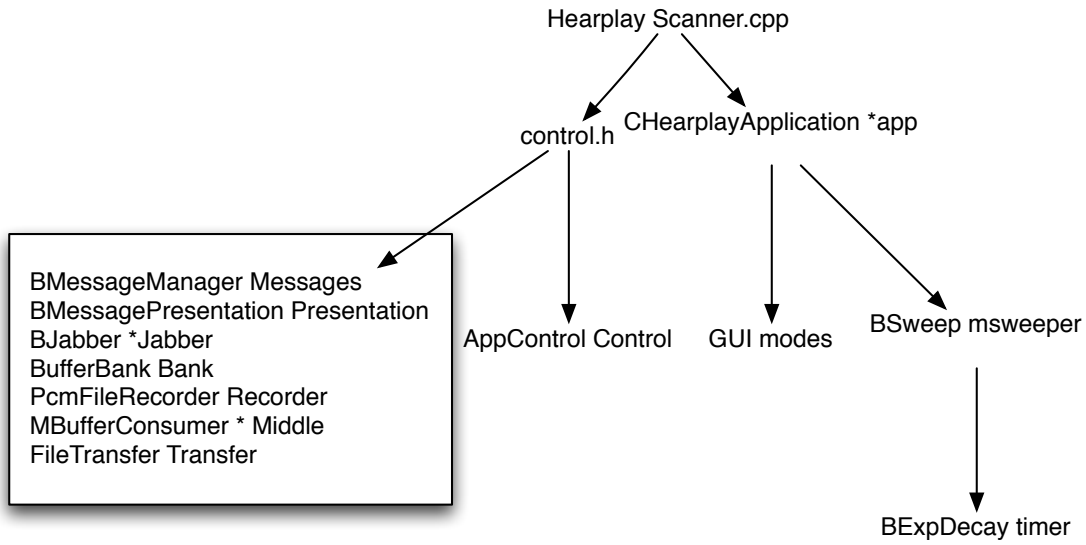


Figure 4-7: System architecture for Hearplay Scanner program

ganizes the sweep, interacting with each channel that is active and passing control to `BMessagePresentation`.

### Controller

Perhaps the central class of Hearplay Scanner Controller is the `BAppController` class, which ties together the Model classes with View Classes. The `BAppController` class itself does not encapsulate program logic. The responsibility of the class is to connect the components of each service. It is also responsible from switching from one user interface mode to another. Since `libsigcce` allows one-to-many connections, this allows several components to respond in a coordinated manner to a system event (such as an incoming synchronous audio burst), fulfilling the design goal of loose coupling.

### Libraries

There were several third-party libraries used to create the Hearplay Scanner program, identified in Table 4.4

Table 4.4: Third party libraries used in Hearplay Scanner

| **Library** | **Purpose)** |
|---|---|
| `expat` | XML processing |
| `GapiDraw` (version 2.02) | Audio playback |
| `libsigcce` | Messaging between objects |
| `portlib` | Utility library, including console |
| `sqlite` | Database program |
| `jabberooce` | Jabber client library |

## 4.6 Comparison to related work

As discussed previously, audio information and services have been aggregated on the server side. Though different from the server-side service aggregation approach of Impromptu, Hearplay uses networked handheld iPaqs on a 802.11 wi-fi network to provide audio services [24]

TattleTrail, implemented on the Impromptu architecture described in Chapter 2, is a server-side approach to aggregation of audio services and alerting [28]. In TattleTrail, synchronous communications are recorded on the server for later and can be browsed asynchronously, sped up on the device, so that one can catch up on conversation that was missed previously. The TattleTrail system operates in three modes: awareness, synchronous mode, or asynchronous mode. The system uses the speech recognition facility of Impromptu to switch from one mode to another. Hearplay, by contrast, is always in the awareness mode, but attempts to batch asynchronous audio for playback at intermediate times. Hearplay is both more specialized and less powerful in some ways than Impromptu, but could be useful in situations where awareness is the primary concern.

SimPhony, also implemented on the iPaq, uses the observed frequency of asynchronous person-to-person messaging to automatically transition users to synchronous chats [20]. Hearplay is a bit similar in using the timing of recieved events to control the frequency of audio presentation. However, Hearplay does not address handoff from asynchronous to synchronous audio.

Hearplay is unique from these systems in that it relies on the use of audio as a means of annotating mobile content. Like Simphony, it relies on application-level decisions on

how to handle incoming events, instead of the server-side approach of TattleTrail and Impromptu. The system progressed to the point where the user interface could be deployed and demonstrated, but the system was not deployed for a user group.

## 4.7 Critique

Hearplay was primarily an effort to apply audio scanning to the problem of distribution, aggregation and presentation of audio annotated media to support group interaction. The broad scope and technology-driven approach of Hearplay raised many questions during the course of implementation, some of which detracted from the purpose of testing the audio scanning user interface metaphor. Several criticisms can be raised which impact what lessons can be drawn from the system.

Asynchronous audio itself was in the early stages of commercialization and familiarity when Hearplay was created. The term *podcasting* had not yet been created, although there were experiments in distributing asynchronous audio over the internet since the late 1990s. In Hearplay, the relative lack of user familiarity with stored audio on the mobile complicated the attempt to aggregate this audio with synchronous communications on the mobile device.

In a deeper respect, Hearplay anticipated but failed to define or explore the consequences of user-generated and shared content. On top of this, it did so in both audio and mobile media, both aspects of which would only be commercially explored much later, in 2007 and onwards [29]. The early implementation of Hearplay was created in 2001 - several years before the commercialization of social media and user-generated content. In particular, the early social media and groupware aspects of Hearplay complicated the creation, testing, and evaluation of the audio scanning user interface. When Hearplay was created, social network sites were in the early stages of development. The early failing of sites like SixDegrees, which closed in 2000, shortly before Hearplay was created, was that early adopters complained there was little to do with the site than create and respond to friend requests [29]. I believe that the time was just not right for understanding what audio scanning could add – above and beyond the utility of other systems for social media. Given the acceptance of podcasting, social media sites, and media consumption on the mobile, perhaps future work

could revisit these questions.

Another failing of the Hearplay design process was that the technology was not created in situ amongst a group of users with an expressed need for this kind of always-on information appliance (as Lakshmipathy successfully did, for example, in the Simphony project) [20]. As the primary medium of Hearplay was shared audio, it was difficult to find a group to use (especially outside the laboratory) to evaluate the design.

To evaluate audio scanning, it would therefore be necessary to separate the groupware and social media aspects of Hearplay from the evaluation of the audio interface itself. Although in this respect the design goal of Hearplay was unfulfilled, the implementation of Hearplay allowed iterative approach to the design of the audio scanning interface metaphor. The second implementation of audio scanning, Hearwell, used this as a guiding principle to focus on the question of how to constrain the entailments of the user interface in a use case with narrower scope.

# Chapter 5

# Hearwell

This chapter describes the application of audio scanning in a system called Hearwell, to demonstrate how the availability of stored audio and its timely playback can support certain kinds of health and wellness goals with mobile devices.

In previous chapters, I have discussed the utility of the radio scanner metaphor in support of shared mobile audio annotations. Related work has shown that new interaction patterns on the mobile involve continuous partial awareness, fluid organization of tasks, and micro-coordination between social participants. The Hearplay project demonstrated how these patterns could relate to the task of sharing audio annotated data among a distributed workgroup. The main critique of Hearplay was that focussing first on the social media use case in retrospect complicated the task of assessment of the new user interface metaphor.

One motivation for Hearwell as a follow-on project is to better isolate the ways in which audio scanning could support continuous partial awareness and fluid organization of tasks. In this use case, the goal is to provide users with a frequent, lightweight, and rapid way of reviewing information as a prerequisite for taking action to maintain a healthy lifestyle. As in Hearplay, the audio scanning interface metaphor has been used in the implementation of this functionality.

This chapter is divided into four sections:

- The first section describes the design motivations in detail.
- The second section outlines the specific manner in which the audio scanning interface

metaphor was applied.

- The third section describes the technical implementation details of Hearwell system.
- The final section relates the system to previous work and presents a limited-scope usage evaluation of the system.

## 5.1 Motivations

### 5.1.1 Need for a ubiquitous health information environment

New media applications are expanding the ways in which healthcare information can be delivered with mobile technology. For mobile devices, health and wellness applications are a popular category of application. In the niche of audio media, health and wellness podcasts are common.

Most of these applications, however, enable the user to actively search for or record information. A mobile user must launch an application to complete a health-related task, or actively download or search for new content. If relevant health information could be delivered at appropriate moments in daily life, the mobile could create the potential for a more continuous awareness, setting the stage for fluid organization of actions to achieve those goals.

Research has shown that chronic disease patients must employ varying modes of information seeking for medical information: an active mode where there is certain information that is being sought after, or passive awareness that comes from monitoring information that is potentially beneficial [30]. Much of what we learn comes through us through passive undirected behavior, and this can produce a valuable feeling of empowerment as we learn about conditions that directly affect our health [31, 32].

### 5.1.2 Need to support spontaneous learning

The often emotionally-charged, sometimes disconcerting nature of health information creates a need for more spontaneous learning strategies. If a user is emotionally upset or concerned over a medical condition, information seeking and effective action will be more

difficult. However, if opportunities for learning about ways to keep healthy or relevant information about a medical condition are interspersed in the background interaction with information in daily life, discovery of relevant information could provide a subtle but noticeable benefit. In creating more opportunities for micro-interactions and reinforcement, the user would be more likely in an emotional state that would reinforce the application of domain knowledge in daily life.

### 5.1.3 Need for reminder of health goals and lightweight reinforcement of health knowledge

The focus of this thesis is on a new audio user interface metaphor for mobility. Though research into the kind of interventions needed to support health behaviors is more thoroughly addressed by healthcare professionals and researchers seeking to apply more complete persuasive computing paradigms, I believe that supporting memory for healthcare information is a crucial prerequisite for improving the user experience of health and wellness applications.

Studies have shown that 40-80% of medical information provided by healthcare providers to a patient is forgotten immediately [33]. Memory loss is also associated with the aging process, which compounds the problems for many. If patients cannot remember the basic medical facts of their condition, it is difficult to adhere to treatment or make well-informed medical decisions. A person who feels knowledgeable about health and wellness issues can gain a feeling of control that can make adherence to medical direction more likely.

The application of audio scanning to reinforce knowledge of health-related topics is not limited to chronic disease management. By forming health habits in daily life, small actions can lead to important outcomes. Keeping aware of health and wellness goals can also make a difference in health outcomes over the long term.

## 5.2 Audio scanning

The primary goal of audio scanning is to provide a means for regular playback and review of stored audio related to health and wellness. To support this goal, the implementation of

Table 5.1: Applying entailments of Radio Scanner interface metaphor in Hearwell

| **User Interface Metaphor Entailment** | **Applies to** |
|---|---|
| A radio scanner plays audio in 'bursts'. | Applied in Hearwell. The 'burst' is the fundamental message in Hearwell. |
| A radio scanner plays audio in 'channels'. | There is only one content type, therefore only one channel required in Hearwell. |
| A single button turns a single channel on or off in a radio scanner. | Not applied in Hearwell. Users interact with two GUI buttons for navigation. |
| Content can only be in one channel at a time in the distribution system. | As there is only one channel, this is a truism in Hearwell. |
| There are channel sweeps in the distribution system. | Applied in Hearwell |
| The regularity of channel sweeps can be chosen in the distribution system. | Not applied in Hearwell. The user can opt into channel sweeps after a phone call is completed, when attention is drawn to the device. |
| There is a squelch knob in the content distribution system. | Not applied in Hearwell. |

audio scanning in Hearwell differs in several respects from the implementation in Hearplay.

The following table summarizes the entailments of audio scanning and their application in Hearwell. The remainder of this section will discuss several entailments in greater detail to more fully describe the desired user experience, as well as to compare it to the Hearplay system.

The three fundamental entailments of audio scanning also translate directly into entities in Hearwell: bursts, channels, and sweeps. The entity diagram of Figure 4.2, showing the application of entities in Hearplay, applies equally for Hearwell. In fact, looking at this aspect of the system, Hearwell can be thought of as simply a distinct channel within Hearplay to support a particular usage for review of health and wellness information relevant to daily life.

### 5.2.1 Bursts

As in Hearplay, the scanning interface in Hearwell plays audio in bursts.

Burst types in Hearwell are more constrained than in Hearplay. A burst in Hearwell

always consists of a segment of stored audio and a short text (an indication of subject, purpose, a question, or other annotation) that can be associated with the content of a particular audio segment.

Bursts in Hearwell can be pre-defined in a data specification called the cue sheet format. Cue sheets are metadata files that define how the audio tracks of a CD are laid out, delineating the start and stop time index of each desired track. This metadata files are often used when multiple audio pieces are combined in a single file, avoiding the need to separate the single file into separate pieces. Cue sheets can be generated by many audio processing tools.

For Hearwell, little modification was needed to the syntax of a cue file to support audio bursts. An example of a cue file is shown below. The cue file below describes 4 bursts, one containing an entire mp3 file, `F01.mp3`, and 3 shorter bursts which are in fact portions of the larger file. The `TITLE` metadata in the cue file format has been used to define a short piece of text to be displayed by the Hearwell program when the burst is played, and the `INDEX` metadata is used to mark the time index of the beginning of the burst. An empty string in the `TITLE` metadata is used to end a burst when two consecutive bursts do not share a boundary.

---

```
TITLE "Tips for healthy eating when you go out"
   FILE "F01.mp3" MP3
      TRACK 01 AUDIO
         TITLE "Don't be afraid to special order when eating out."
         INDEX 01 00:52:45
      TRACK 02 AUDIO
         TITLE ""
         INDEX 01 01:00:89
      TRACK 03 AUDIO
         TITLE "Look up the menu before you go to eat out."
         INDEX 01 01:11:44
      TRACK 04 AUDIO
```

```
    TITLE ""
    INDEX 01 01:18:24
TRACK 05 AUDIO
    TITLE "Don't drink cocktails when you go out."
    INDEX 01 01:36:39
TRACK 06 AUDIO
    TITLE ""
    INDEX 01 01:44:93
```

---

The cue sheet format adaptation used in Hearwell has certain drawbacks. It is inefficient in the use of the empty `TITLE` metadata tag. Multiple levels of nesting would be useful to describe more structured content. Also, the format does not allow concatenation of separate segments into a single burst. For these reasons, future work could investigate the use of XML based format to provide structured audio metadata for applications like Hearwell. Nevertheless, the cue sheet format is easily created by existing audio tools, easily modified by hand if necessary, and quite readable. The cue sheet format was adequate to the task of defining bursts in the Hearwell implementation.

### 5.2.2 Channels

Only one channel is used in the Hearwell implementation, since bursts are used from the same source for the same purposes. However, the possibility of separate channels in the entity model allows flexibility in aggregating content of different kinds. Hearwell could aggregate audio bursts to be reviewed in a range of relevant subjects to be presented in one sweep, or combine these bursts with other types of relevant information, as was explored in Hearplay (calendar reminders or communications, for example).

### 5.2.3 Sweeps

Bursts of audio are played back during a sweep, as in Hearplay. The structure of the sweep is similar to Hearplay, with the sweep beginning and ending with an audio icon. As there is

currently only one channel, no audio icon is used to divide channel presentation. Differences from Hearplay include the following:

- A sweep in Hearwell begins not at a preset time, but at the time of a triggering event. In the current implementation, the triggering event is the end of an outgoing phone call. The purpose of this is to present content when user attention has been drawn to the device. It also merges synchronous communication with asynchronous audio playback in a way that was not explored in Hearplay. Other triggering events that could be explored in future work include closing or opening an audio program (the music player for example) or entering or leaving a defined location.

- A sweep in Hearwell does not begin unless the user opts in. A dialogue button appears, asking the user if review of previously heard material should start. This is to support create anticipation and to encourage the user to precommit to the review task, ensuring more fluidity in switching to the review task, since the entering the review sweep is completely volitional.

- A limited number of bursts are included in the scan. The purpose of this is to keep overall playback time short, to create regularity and to re-enforce expectations of how the interface operates, which could prove to increase user acceptance. A sweep consists of familiar bursts for review and unfamiliar bursts for presentation of new information. The channel will select particular items for the sweep as described in the Asynchronous Presentation Strategy section below.

- The user does not delete bursts from a sweep, but still can skip ahead at any time. Skipping ahead, however, can change the order in which bursts are reviewed in later sweeps.

- Though a burst can be repeated at a later sweep, Hearwell introduces the idea of dependencies between bursts to support review. At the initial sweep, the Hearwell user will listen to a complete audio clip (of up to perhaps 3 to 4 minutes in length). If the user skips ahead on this presentation, the position is saved and will resume in the next sweep. After listening to the complete audio clip, the original burst is removed

from the channel, and the review bursts will added for the purpose of review. In general, the review bursts will be shorter segments of the original burst. This allows import points to be emphasized and minimizing time for review.

### 5.2.4 Asynchronous presentation strategy

As in Hearplay, a timing heuristic is used in Hearwell to select which content should be included in the sweep. For the purposes of mnemonic review, the use of a back-off heuristic is a widely-used technique, often referred to as spaced repetition. Spaced repetition is particularly associated with Paul Pimsleur, who tested a method of graduated interval recall as early as 1967. The Pimsleur strategy attempts to combine a variety of behavioral influences, including immediate reward, combining old and new material, and gradual but steady reinforcement of learning to the task of language learning [34]. Similar strategies have been used in a variety of computer-assisted learning programs, and is also commonly used in audio material for language review. Academic research supports the claim of benefits of spaced repetition strategy, however, a direct comparison of acquisition schedules and their effect on long-term retention is largely lacking[35]. Instead, Hearwell attempts to show that the opportunities for engaging in review can be expanded with the use of mobile devices, no matter what the acquisition schedule. Audio scanning is user interface metaphor that allows such use cases to be easily integrated with other forms of both asynchronous and synchronous audio communication for the purpose of organic learning in daily life.

As entering a sweep in Hearwell is volitional, the asynchronous presentation strategy entails something of a tradeoff. On one hand, there is no easy way without breaking the model to ensure that a burst is played at some particular point in the future. On the other hand, when the burst is played, we have greater if not absolute certainty if the burst was heard by the user. To compensate for volitional strategies, Hearwell implements an upper and lower bound instead of a target presentation time that is used to trigger Hearplay sweep playback.

To describe the presentation strategy in Hearwell, consider the following acquisition schedule in Table 5.2. The first column refers to the number of times a burst has been played. The target is an optimum time for playback after the last time a burst was presented

Table 5.2: Burst presentation schedule in Hearwell

| **Repetition** | **Target (days)** | **Lower** | **Upper** |
|---|---|---|---|
| 1 | 1 | 0 | 2 |
| 2 | 3 | 1 | 4 |
| 3 | 5 | 3.5 | 6.5 |
| 4 | 8 | 6 | 10 |
| 5 | 12 | 8 | 16 |
| 6 and up | 20 | 15 | 25 |

in a sweep. However, a burst can also be presented at any time between the lower and upper time bounds, which are set around the target playback time. As the time window for presentation increases as the number of repetitions increases, the presentation strategy should attempt to favor presentation of bursts with fewer repetitions over bursts with more repetitions.

The heuristic then looks at the time since last presentation. The following considerations are taken in Hearwell.

- Bursts that have not been reviewed, even when the upper bound has been exceeded, need to have priority. The strategy that Hearwell uses is to add these bursts to the current sweep, but not to count the presentation as a repetition. More stringent approaches would be to decrement the repetition count, or even zero out the repetition count completely. It is entirely possible, of course, that a user will not enter a sweep at all, causing all review material to become stale. This special case is not considered in this implementation. The first strategy that should be used in this case is to increase the opportunities for the user to opt-in to a sweep, driven perhaps by the amount of data for review in the system.

- Bursts that have not been reviewed before the earlier time limit can be ignored.

- For bursts that fall between both earlier and later time limit, try to add these timely bursts to the sweep until the sweep length has been reached. As items that have been skipped over in the past can be assumed to have been more familiar to the user, the presentation strategy should first add those items that have not been skipped in the

past. More complicated strategies are possible: if the system tracks the percentage of times the item has been skipped in the past, or what percentage of content in the burst the user reviews before skipping ahead, these can be used to provide a more finely attuned ordering strategy. As in the precise acquisition schedule, however, determining which sort of strategy is most effective would require future experimentation and long-term user feedback.

The sweep then continues until all bursts within the time window have been added, or the limit in sweep lengths has been reached. If the sweep limit length has not been reached, Hearwell will add new material to the end of the sweep. A flowchart gives an overview of the burst selection process. The desired effect is to create a sweep in which:

- Newer bursts for review is presented before older material for review.
- Bursts are reviewed within a predetermined time period.
- Only a certain number of bursts are presented in a sweep.
- Bursts are reviewed before unfamiliar material is presented.

### 5.2.5 User interaction

Interaction in the Hearwell use-case, as implemented, is outlined in Figure 5-2. A user will first dial a phone call normally. Before the call is placed, a screen will ask the user to review audio content after the call. The call is then placed normally. After the call, if the user has opted-in, the sweep is created and bursts are presented according to the strategies described above.

To summarize, Hearwell focuses on a more constrained use case that differs from Hearplay in several ways.

- Hearplay is a social information appliance whereas Hearwell is designed for individual consumption and review of audio.
- Hearplay is designed to aggregate content, whereas Hearwell is designed to present information of one focused type.

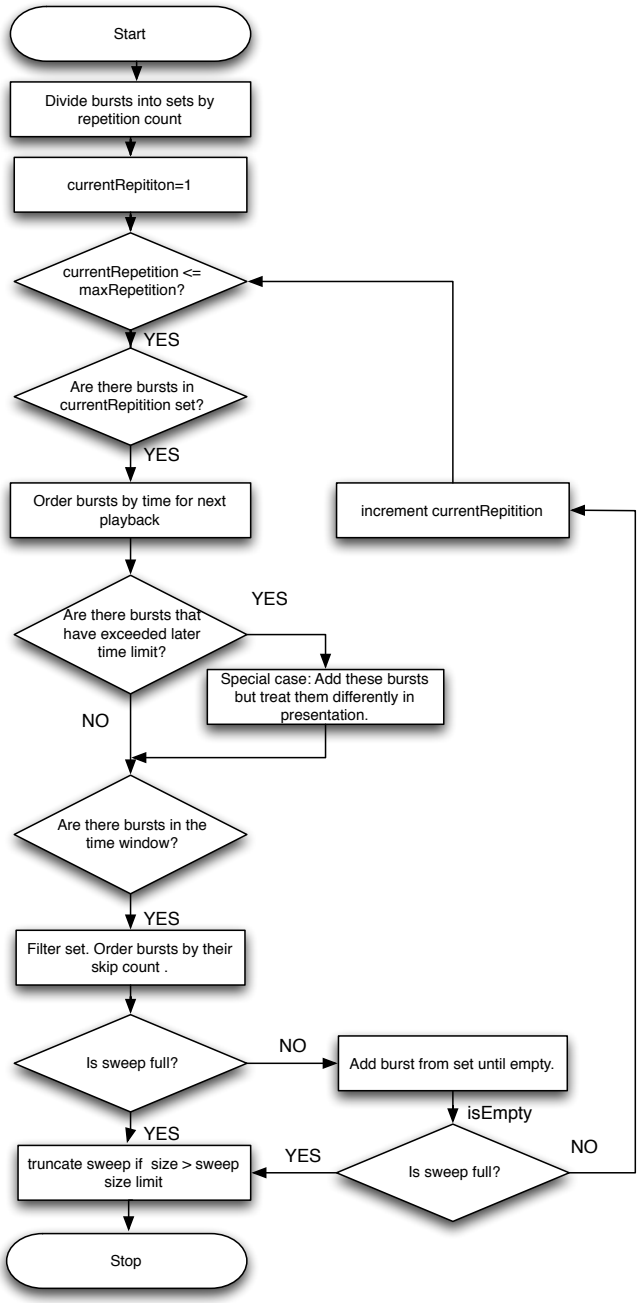


Figure 5-1: Flowchart for burst selection in Hearwell sweep construction

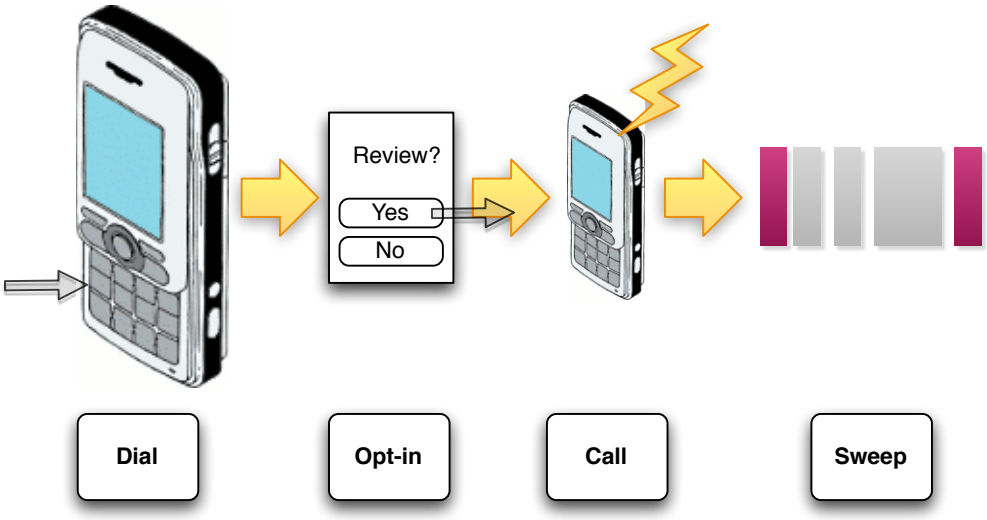


Figure 5-2: Interaction in Hearwell use case

- Hearplay uses a presentation backoff strategy in which stale content can be pushed further back, whereas the Hearwell presentation strategy is intended to allow review in a proscribed window of time.

- Hearplay does not tie asynchronous audio content presentation with the opportunity for synchronous communication, whereas Hearwell uses the expensive process (in terms of time and attention) of engaging in synchronous phone conversation for spaced repetition of relevant content.

## 5.3 System

### 5.3.1 Architecture

Hearwell was programmed in Java on the Android smartphone. Hearwell was intended to be used on an intermittent basis, supporting the process of familiarization and review of stored audio content. The system is composed of only two programs, running on the cellphone client: a program that plays back an audio sweep to the user composed of audio material for review, and a client monitor that launches at boot-time to capture changes in phone state for presentation and logging purposes.

### Scanner

The scanner program presents a sweep of audio bursts. Unlike Hearplay, which initiated a sweep when the age of the material in a channel demanded it, Hearwell enters a sweep once a triggering event occurs. As the attention of the user is fully occupied when placing a call, this is used as a point for giving the user the opportunity to indicate whether a sweep after the call ends.

The Android platform provides the ability for programs to register to receive system-wide events, or *intents*. The Hearwell scanner uses this capability to register to capture the *android.intent.action.NEW_OUTGOING_CALL* intent. This intent is broadcast by the dialer directly. By intercepting this intent, the Hearplay scanner can present a dialogue screen to the user. After the user provides input, an API call is used to call the number directly.

The scanner itself contains classes to read cue sheet information associated with bursts and state information stored in a client-side database, to construct a sweep according to the strategy identified above, to play audio events, and to capture GUI interaction to allow the user to navigate between bursts. Compared to Hearplay, the implementation is much more compact, since there is no need to provide communications or file transfers directly.

A complication for implementation of Hearwell on the Android platform was that programs cannot register for an intent to indicate when a call has ended, which is needed for the scanner to initiate a sweep after the call. The solution to this problem was to create a separate monitor service that starts at time of device boot, and sends a customized intent to the scanner when the end of call is detected.

### Client monitor

In the Android platform, it is possible to read the telephony system context programmatically. This strategy was used to monitor the state of the phone as shown in Figure 5.1.

To provide an *end call* event, a service was created that polls the telephony service to read the current system context. There are three states that are captured: *CALL_STATE_RINGING*, *CALL_STATE_OFFHOOK*, and *CALL_STATE_IDLE*. After launching the service, the telephony service is polled at the frequency of once per second. A finite state machine model is

used to identify the state of the phone. When the service detects a *CALL_STATE_RINGING*, for example, the model transitions from *IDLE* to *OUTGOING CALL*. This allows discrimination between incoming and outgoing phone call states. Once the service returns *CALL_STATE_IDLE* from these in-call states, an application intent is broadcast for the scanner to intercept. This implementation proves effective, though there is the potential of a marginally greater battery drain due to the background polling process. The first implementation of Hearwell also used this procedure to capture outgoing phone call events. This proved to be inferior in performance to a strategy which registered to receive the system-provided *android.intent.action.NEW_OUTGOING_CALL* intent, however. If the Android telephony program were to be modified to allow events to be broadcast directly, this workaround would be rendered unnecessary.

For use in testing and evaluation, the client monitor also captures context information, including currently running programs on the device, and changes in cell ID. The purpose of capturing this information is to capture context for use in post-usage interviews with test users.

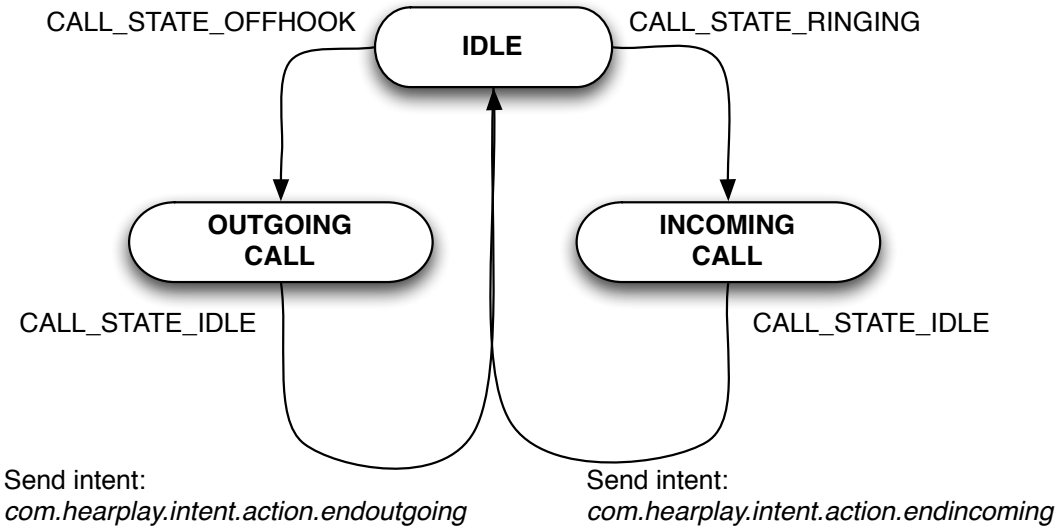


Figure 5-3: State machine diagram for client phone monitor

### Server side

As the purpose for Hearwell did not involve sharing media or live distribution of content, there was no server implementation necessary. Nevertheless, future work could support discovery and distribution of content. The minimum requirements that Hearwell introduces are discovery of audio content and associated cue sheets files. Cue sheets themselves are a content type that are easily supported on web servers as a specialized text based content-type.

## 5.4 Evaluation

### Overview

To evaluate the current implementation of Hearwell, my primary goal was to identify particular design factors that most impact the user experience in interaction with the system. I used a combination of user interviews before and after a period of use of the system. This test period ranged from 1 to 4 days. Before the test, I interviewed participants to allow them to self-assess their interest in nutrition and exercise. For the testing period, I gave them a phone provisioned with Hearwell. Content bursts during the test were based on segments of a public radio podcast of an 10 minute interview with a nutrition expert, evaluating common nutrition beliefs. Each burst consisted of a question and answer addressing a particular topic. In general, the short dialogue question-and-answer format produced useful segments of 20 seconds to a minute in length. Examples of topics include the question of whether or not stress leads to weight gain, or how much weight loss is safe per week. One participant, who is a competitive runner, also received segments from an interview with a running coach about interval training. After using the system for a number of days, I asked the participants to step through a day of usage and give me their impressions of how the system behaved, as well as their qualitative impressions of listening to nutrition information delivered after a phone call. I asked them to identify aspects of the design they felt was not workable or could be improved. By using a series of interviews, I attempted to ground the evaluation of the system in the daily activities of the participants.

Three participants were recruited from the Cambridge community. These testers were of

approximately the same age, and were self-selecting for their interest in health and fitness. Only one of the testers had a technical background. The participants expressed an interest in nutrition, exercise, and wellness. They were also perhaps more fit and health-conscious than the general population. Though the number of participants was too small to reflect a representative sample of mobile users, the range of interests in health and wellness as well as the range of familiarity with podcasts and stored audio allowed a degree of design introspection. The users included:

- a 48 year old competitive runner (male), training for a series of 5k runs. He runs 25-28 miles per week. He actively reads two running magazines, and follows online sites and blogs to learn training tips. He identified his concerns as nutrition before and after a competitive event, timing of when to eat and what, what amounts of water to drink, and setting up training details, including how much to run and when.

- a 40 year old project manager, mother of two daughters. She exercises somewhat frequently. She said she gains motivation in her efforts when she sees celebrities on TV who have been working out, and enjoys listening as well to talk shows that focus on diet and nutrition.

- a 45 year old manager (male), who uses podcasts and listened to radio frequently. He makes an effort to exercise regularly, and expressed the belief that he knows a little about nutrition, but at the same time recognizes a need to learn more and to take action on what he does know. He makes on average between 5-15 calls a day.

**Results and design implications**

Several themes emerged during this brief trial. The first and strongest theme was that all users, perhaps not surprisingly in retrospect, had similar usage patterns in *when* they listened to audio. All three of these users used the system in the car. This context is interesting as a potential for health and wellness education; it creates a regular time of transition as well as a private time for reflection and planning while in transit. The 40 year old project manager indicated that listening to exercise information, even if it wasn't immediately actionable, put her in mind of thinking about what she would be doing when

she reached the gym. The competitive runner, similarly, made most of his calls in the evening but ran in the mornings. The training information he reviewed in the evening, however, allowed him to mentally prepare for things he wanted to accomplish in his next training session, and reminded him of other preparations on evenings before race days (what he should eat before the race, for example)

The second theme was that the users were quite willing to listen to short content after a phone call, and that the repetition of content itself was not identified as a problem, at least for the limited time under testing. Longer testing periods would be necessary to determine the best schedule for content repetition for intervention and reinforcement. The conversational format fit in well after a phone call. The competitive runner was most vocal in support of repetition, and said that he could listen to some tips "even a hundred times" without the content getting old. He said he actually remembered reading the article the source podcast was based on, and that the audio format was better because it went into a bit more depth. He also enjoyed listening to the spoken interaction with the coach. This training content was the most specialized, so perhaps the amount of information that must be mastered, combined with the the conversational format, increases the value of repetition.

However, two users indicated that they wanted to hear longer and more complete audio content in several contexts: while driving, at the office, or at the gym. The design that was suggested by one user was to use the spatial orientation of the device to change playback modes. The suggestion was that the device should play a sweep only while held at the ear. By setting the device down during playback, longer content associated with the burst could then be played in full. A similar idea from a different user was to use the bursts to select longer content to be played on the radio, with its more familiar and convenient controls. This would also allow the phone to be freed for further usage. In both of these scenarios, the short burst is very workable as a 'preview' of a longer audio piece. This suggests an overlooked usage of audio scanning - information-seeking and browsing within a set of asynchronous audio content while multitasking on the move.

A third critical theme was that more audio structure was needed to make the structure of the sweep more apparent. More controls over playback were desired as well. The lack of sound cues before the call was confusing, because users forgot to interact with a dialogue

screen before the call. In Hearwell, unlike Hearplay, audio within the same channel was not separated by an audio icon. As there is little silence between these bursts, the audio experience ran together for all users, especially audio from the same source. The 40 year old project manager remarked: "to tell you the truth, it was like listening to the radio." This is a fair observation of the current user experience of Hearwell without disambiguation within a channel. The competitive runner was also confused, thinking that the several bursts he had heard where in fact one piece of audio. This effect was compounded because both of these users listened to the audio with the phone to their ear, hiding the screen that displayed content information. The project manager desired the same set of controls familiar on music player (stop, pause, forward, and back). The lesson learned on this point is that audio icons and familiar controls are needed at every transition point in this interface, even within a single channel.

Finally, for some users, more control over context of playback is highly important, and can make some bursts more actionable or relevant at different times. One user even suggested that he would like to receive a notification if no call was made for a period of time (reminiscent of the Hearplay implementation). For the competitive runner, faced with the requirements of regularity in training schedule and the changing demands of daily life, a finer degree of control over when bursts should be presented was important. The manager wanted to support a more active learning mode, where the content would be presented in a different way after hearing the burst a few times. He suggested that a health tip might be rephrased into a question. Supporting these interactions would require more meta-information about the content of each burst, and perhaps a much more complicated interface to support user preferences. Content tagging, for example, could help users collaboratively organize information for this kind of contextual filtering. These usages would also require more internal structured data for each burst, as well as specification of the relationships between bursts themselves.

### 5.4.1 Comparison to related work

Hearwell can be situated between two classes of mobile applications: applications that assist human memory, and applications that attempt to intervene in daily life to create and sup-

port behavioral change. Computer-assisted learning tools have already been mentioned in the section on spaced repetition, and content- and context-triggered reinforcement has been used in systems such as *Remembrance Agent* and *What Was I Thinking?* to provide memory triggers in daily life [36, 37]. The role of acquisition schedules, as discussed previously, is not completely understood. The wider context for memory reinforcement might prove to be as impactful as the time schedule itself - and mobile devices that monitor and record broad context information could support discovery of the linkages between remembrance and daily life. An interesting avenue for further research in the health domain might be to pursue related work in context-triggered reinforcement as part of the interaction of such systems, so that mobile systems could serve dual usage as both real-time memory prostheses and support development of natural mnemonic potential [38].

Persuasive technologies, especially those for healthcare applications, expand beyond the need for health instruction and reinforcement of knowledge to exploring how to leverage the opportunity for behavioral intervention. B.J. Fogg, in a review of mobile health applications, identifies two factors that contribute to the mobile's success in this area [39]. First, the mobile's convenience factor makes it possible to rehearse desired activity in the middle of the real world. Secondly, the mobile is able to intervene at opportune moments for influence because it is always available and responsive. In a similar spirit, Hearwell is built on the intuition that audio review can be less obtrusive when it is organized around situations of audio communication.

Audio can provide a supportive means of interaction in these use cases. Many so-called "mobile-health" systems are not sufficient - adapting web content for the mobile device, for example, creates problems for low-literacy clients [40]. By providing health information without the user explicitly initiating an interaction, health information could be presented in less stressful situations of daily life. The goal of Hearwell, similar to such systems as *ViTo*, is to raise user awareness and to overlay health information over other patterns of media consumption [41]. Hearwell, if accepted, would support an argument that the media of persuasion on the mobile can be an important factor. As argued in the introduction, audio could prove be more emotionally reassuring and less distracting in daily life, especially when interaction is not initiated by the user. However, supporting these audio interactions in an

unobtrusive manner is only a first step to creating persuasive systems. Identifying relevant opportunities and means for behavioral intervention becomes an intrinsically fundamental research question, beyond the question of user interface design.

# Chapter 6

# Conclusions

Unlike the graphic user interface, which is tightly constrained by the device manufacturer, current mobile platforms leave much of the design of the content of the mobile audio interface and of the audio experience to the application programmer.

This freedom is a radical change in some respects. Since audio was first recorded until the digital revolution, audio technology usage was constrained by the real-world characteristics of the audio recording itself. A phonograph player, for example, could not be easily transported, and could not be played while in motion. The tape recorder, however, proved to be quite portable. But there was a tradeoff - a listener could skip easily from one song on the phonograph to one much later. This was difficult to do with precision on a cassette tape recorder. While we may expect that new technology will simply replace older technology, from a look at the past we observe that new audio technologies have supplemented existing systems by allowing utilization of a different set of affordances.

With the growth of digital technology, this has changed. Certainly our mobile devices in 2009 are much more capable of handling audio as a datatype in both synchronous and asynchronous forms, often in a single device. But often we bring the same user interface metaphors and distinct applications to new technology, even when old physical constraints no longer apply. Change in usages of course is incremental and lags behind technical progress, and new usages only emerge as users become familiar and exploit new possibilities of interaction.

At some point, however, older user interface metaphors may seem less appropriate for

how technology is actually used. It is interesting that for the mobile phone, in its role as an all-purpose digital audio appliance, this disruption may be caused by mobility changing the patterns for social interaction. Mobility creates the opportunity for application of new user interface metaphors; either created anew or appropriated from sources that seem more relevant for a given usage domain.

In this thesis, I wanted to show that audio scanning is one such possible appropriation. The audio scanning interface metaphor provides a lightweight and flexible user interface metaphor for the use of audio in mobile applications. To explore this topic, I focused on two use cases that required a mix of micro-coordination, fluid organization of tasks, and continuous partial awareness.

Hearplay was an effort to combine audio communication forms into a mobile system for sharing of audio annotations. The organization of aggregated channels with varying characteristics and usages was the primary challenge, and the audio scanning interface adapted well to this usage, although the social usages of the system were unexplored.

Hearwell, on the other hand, applied the metaphor to the task of reinforcing organic learning for health and wellness. The insight of Hearwell was that interaction with one function on the mobile phone can be used as an opportunity to review content, tying synchronous communication and consumption of stored audio together.

Both implementations relied on the same fundamental entailments of the audio scanning metaphor. Each applied a back-off timing strategy to the domain to organize the presentation of audio content. My hope is that, considered together, this work points to ways in which the interface metaphor of audio scanning can support the interrelated facets of mobile interaction in a range of usages.

# Bibliography

[1] D. A. Cieslikowski, "Key trends in ict development," in *Information and Communications for Development 2009: Extending Reach and Increasing Impact*. Washington, D.C: World Bank, 2009, pp. 125–155.

[2] Press release: Worldwide mobile cellular subscribers to reach 4 billion mark late 2008. International Telecommunications Union. [Online]. Available: http://www.itu.int/newsroom/press_releases/2008/29.html

[3] Press release: First quarter semiconductor sales up 7.3 percent over 2005. Semiconductor Industry Association. [Online]. Available: http://www.sia-online.org/cs/papers_publications/press_release_detail?pressrelease.id=29

[4] D. Morton, *Off the Record: The Technology and Culture of Sound Recording in America*. Piscataway, N.J: Rutgers University Press, 1999.

[5] S. Plant. on the mobile: the effects of mobile telephones on social and individual life. Motorola. [Online]. Available: http://www.motorola.com/mot/doc/0/234_MotDoc.pdf

[6] M. Ito. (2004) Personal portable pedestrian: Lessons from japanese mobile phone use. Paper presented at Mobile Communication and Social Change, the 2004 International Conference on Mobile Communication in Seoul, Korea, October 18-19, 2004. [Online]. Available: http://www.itofisher.com/mito/archives/ito.ppp.pdf

[7] S. Farnham and P. Keyani, "Swarm: Hyper awareness, micro coordination, and smart convergence through mobile group text messaging," in *HICSS '06: Proceedings of the*

*39th Annual Hawaii International Conference on System Sciences*, vol. 3. Washington, DC: IEEE Computer Society, 2006, p. 59a.

[8] R. Ling and B. Yttri, "Nobody sits at home and waits for the telephone to ring: hyper-coordination via mobile phones in Norway," in *Perpetual contact: Mobile communication, private talk, public performance.*, J. Katz and M. Aakhus, Eds. New York, NY: Cambridge University Press, 2002.

[9] L. Stone. Linda stone's thoughts on attention and specifically, continuous partial attention. [Online]. Available: http://www.lindastone.net/

[10] M. Kakihara, "Fluid organizing of work in the ubiquitous information environment," *Designing Ubiquitous Information Environments: Socio-Technical Issues and Challenges*, pp. 183–196.

[11] B. Beurer-Zuellig and M. Meckel, "Smartphones enabling mobile collaboration," in *HICSS '08: Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences.* Washington, DC, USA: IEEE Computer Society, 2008, p. 49.

[12] K. O'Hara, M. Perry, A. Sellen, and B. Brown, "Exploring the relationship between mobile phone and document activity during business travel," in *Wireless world: social and interactional aspects of the mobile age.* New York, NY, USA: Springer-Verlag, 2002, pp. 180–194.

[13] V. Bellotti, B. Dalal, N. Good, P. Flynn, D. G. Bobrow, and N. Ducheneaut, "What a to-do: studies of task management towards the design of a personal task list manager," in *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM Press, 2004, pp. 735–742.

[14] The mobile audio media study. Arbitron. [Online]. Available: www.arbitron.com/downloads/mobile_audio_study.pdf

[15] C. Schmandt, *Voice Communication with Computers: Conversational Systems.* Van Nostrand Reinhold Company, 1993.

[16] E. D. Mynatt, M. Back, R. Want, M. Baer, and J. B. Ellis, "Designing audio aura," in *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1998, pp. 566–573.

[17] R. Hull and J. Reid, "Designing engaging experiences with children and artists," in *Funology: from usability to enjoyment*. Norwell, MA: Kluwer Academic Publishers, 2005, pp. 179–187.

[18] P. Dourish and S. Bly, "Portholes: supporting awareness in a distributed work group," in *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press, 1992, pp. 541–547.

[19] M. S. Ackerman, B. Starr, D. Hindus, and S. D. Mainwaring, "Hanging on the 'wire: a field study of an audio-only media space," *ACM Trans. Comput.-Hum. Interact.*, vol. 4, no. 1, pp. 39–66, March 1997.

[20] V. Lakshmipathy and C. Schmandt, "Simphony — a voice communication tool for distributed workgroups," *BT Technology Journal*, vol. 22, no. 4, pp. 52–61, 2004.

[21] M. I. Graham and K. Karahalios, "Chatamp: Talking with music and text," in *INTERACT*, ser. Lecture Notes in Computer Science, M. F. Costabile and F. Paternò, Eds., vol. 3585. Springer, 2005, pp. 978–981.

[22] R. Kartofel, "Talking in circles: Representing place and situation in an online social environment," M.S. Thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology, Cambridge, MA.

[23] N. Sawhney and C. Schmandt, "Nomadic radio: speech and audio interaction for contextual messaging in nomadic environments," *ACM Trans. Comput.-Hum. Interact.*, vol. 7, no. 3, pp. 353–383, 2000.

[24] C. Schmandt, K. H. Lee, J. Kim, and M. Ackerman, "Impromptu: managing networked audio applications for mobile users," in *MobiSys '04: Proceedings of the 2nd interna-*

*tional conference on Mobile systems, applications, and services.* New York, NY, USA: ACM, 2004, pp. 59–69.

[25] P. Barr, J. Noble, and R. Biddle, "A taxonomy of user interface metaphors." in *Proceedings of SIGCHI-NZ Symposium On Computer-Human Interaction*, 2002.

[26] G. Lakoff and M. Johnson, *Metaphors We Live by.* University of Chicago Press, 2003.

[27] C. Schmandt, "Audio hallway: a virtual acoustic environment for browsing," in *ACM UIST Symposium on User Interface Software and Technology.* ACM Press, 1998, pp. 163–170.

[28] V. Lakshmipathy, C. Schmandt, and N. Marmasse, "Talkback: a conversational answering machine," in *Proc. of UIST '03, Vancouver, Canada*, 2003.

[29] D. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," *Journal of Computer-Mediated Communication, 13(1), article 11.*

[30] D. O. Case, *Looking for Information: A Survey of Research on Information Seeking, Needs, and Behavior.* Academic Press, 2002.

[31] M. Bates. (2002) Toward an integrated model of information seeking and searching. in the 4th int'l conf. on information needs, seeking and use in different contexts (keynote address), september 11-13,2002. [Online]. Available: http://www.gseis.ucla.edu/faculty/bates/articles/info_SeekSearch-i-030329.html

[32] W. Todd and E. Ladon, "Disease management: Maximizing treatment adherence and self-management," *Disease Management & Health Outcomes*, vol. 3, pp. 1–10, 1998.

[33] R. P. C. Kessels, "Patients' memory for medical information," *J R Soc Med*, vol. 96, no. 5, pp. 219–222, 2003. [Online]. Available: http://jrsm.rsmjournals.com

[34] P. Pimsleur, "A memory schedule," *Modern Language Journal*, vol. 51, pp. 73–75, 1967.

[35] D. A. Balota, J. M. Ducheck, and J. M. Logan, "Is expanded retrieval practice a superior form of spaced retrieval? a critical review of the extant literature," in *Foundation*

*of Remembering: Essays in Honor of Henry L. Roediger III.* New York: Psychology Press.

[36] B. J. Rhodes, "The wearable remembrance agent: a system for augmented memory," in *First International Symposium on Wearable Computers, 1997.*, 1997, pp. 123–128.

[37] S. Vemuri, C. Schmandt, W. Bender, S. Tellex, and B. Lassey, "An audio-based personal memory aid," in *UbiComp 2004: Ubiquitous Computing: 6th International Conference, Nottingham, UK, September 7-10, 2004. Proceedings*, 2004, pp. 400–417.

[38] J. Beaudin, S. S. Intille, E. M. Tapia, R. Rockinson, and M. Morris, "Context-sensitive microlearning of foreign language vocabulary on a mobile device," in *Proceedings of the European Ambient Intelligence Conference. vol. LNCS 4794.* Berlin Heidelberg: Springer-Verlag, 2007, pp. 55–72.

[39] B. J. Fogg, *Persuasive Technology: Using Computers to Change What We Think and Do.* Amsterdam: Morgan Kaufmann, 2003.

[40] M. Mackert, B. Love, and P. Whitten, "Patient education on mobile devices: an e-health intervention for low health literate audiences," *Journal of Information Science*, vol. 35, no. 1, pp. 82–93, February 2009.

[41] J. P. Nawyn, "A persuasive television remote control for the promotion of health and well-being," M.S. Thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology, Cambridge, MA, 2005.