

RadioActive:

Enabling large-scale asynchronous audio
discussions on mobile devices

Aaron Zinman

B.S. Cognitive Science
University of California, San Diego
June 2004

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

August 2006

© 2006 Massachusetts Institute of Technology

Author

Aaron Zinman

Program in Media Arts and Sciences

August 11, 2006

Certified by

Judith S. Donath

Associate Professor of Media Arts and Sciences

Thesis Supervisor

Accepted by

Andrew Lippman

Chair, Department Committee on Graduate Students

Program in Media Arts and Sciences

RadioActive:

Enabling large-scale asynchronous audio
discussions on mobile devices

Aaron Zinman

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 in Media Arts and Sciences at the
Massachusetts Institute of Technology

August 2006

Abstract

Current mobile technology works well to connect individuals together at any time or place. However, general focus on one-to-one conversations has overlooked the potential of always-on group and community links. I hypothesize that asynchronous persistent audio is a superior medium to support scalable always-on group communication for mobile devices. To evaluate this claim, one must first have an adequate interaction design before its possible to investigate the qualities and usage patterns over the long-term. This design does not exist for mobile devices. This thesis takes the first step in this direction by creating and evaluating an initial design called RadioActive. RadioActive is a technological and interaction design for persistent mobile audio chat spaces, focusing on the key issue of navigating asynchronous audio. If RadioActive is shown to be a good design in the long-term, I hope to prove with additional studies the value of asynchronous persistent audio.

In this thesis I examine related work, describe RadioActive from a methodologically constrained bottom-up approach, discuss the theoretical rationale behind the design, what seems to work, what doesn't, and suggestions for the future.

Thesis Supervisor: Judith S. Donath
Title: Associate Professor of Media Arts and Sciences

RadioActive:

Enabling large-scale asynchronous audio
discussions on mobile devices

Aaron Zinman

The following people served as
readers for this thesis:

Reader

Chris Schmandt

Principal Research Scientist
MIT Media Laboratory

Reader

Walter Bender

Senior Research Scientist
MIT Media Laboratory

ACKNOWLEDGEMENTS

Thank you Judith, my advisor, for helping guide me through this process, teaching me visual design, and bringing me back down to earth.

Thank you Chris, your audio and practical wisdom greatly helped orient me.

Thank you Walter, you always are willing to provide sage advice and critique on anything both big and small.

Thank you Francis, Christine, Orkan, Fernanda, and Scott. I've always been so impressed with the quality of thought from the Sociable Media Group.

Thank you Sajid, Jon, and the rest of the crazies from the lab. Zero-chi baby!

Thank you Federico for the opportunities, and of course the ham and wine!

Thank you Greg, Erik and Eric, Jonah, Karuna, Gena, Lily, Roger, Bill, Dan, David, Lisa, Diana, Minna, Rhona, and the rest of my friends, for providing support and understanding during my time at MIT. You guys are awesome!

And of course, thank you mom and dad, for selflessly giving me life, opportunity, love, and guidance throughout all these years. I did it!

TABLE OF CONTENTS

1. Introduction	13
2. Related Work	15
2.1 Nishimoto AVM.....	15
2.2 ScanMail.....	16
2.3 Mobile Audio Chat Approaches.....	16
2.4 Browsing Audio.....	17
2.5 Audio Skimming.....	18
3. Designing RadioActive	19
3.1 Design Rationale.....	22
3.1.1 Creating a interface palette.....	22
3.1.2 The chat space.....	23
3.2 Getting to the Current Design.....	25
3.2.1 Message-level view.....	25
3.2.2 Energy-minimizing layout.....	26
3.2.3 Communicating message hierarchy.....	28
3.3 Messages.....	29
3.3.1 Representing audio: message length.....	29
3.3.2 Representing audio: message content.....	31
3.3.3 Other attributes: heard/unheard status.....	32
3.3.4 Other attributes: authorship.....	33
3.3.5 Message interaction.....	33
3.4 Inbox: Compact Thread Representation.....	35
3.5 Other Design Elements.....	37
3.6 Navigating Without Trying.....	37

3.6.1	Transparency in machine-interfaces	38
3.6.2	Linear presentation	38
3.6.3	Choice of coverage	39
4.	Design Theory	40
4.1	Information Navigation	40
4.1.1	Social.....	40
4.1.2	Semantic.....	40
4.1.3	Spatial	41
4.2	Spatial Dynamics	42
4.3	Consequences of Audio	42
5.	Technological Design	44
5.1	Client.....	44
5.2	Server	46
6.	Evaluation	48
6.1	Initial Evaluation.....	48
6.2	eLens	49
6.2.1	System problems	50
6.2.2	Task structure	51
6.2.3	What worked.....	51
6.2.4	Lessons learned	52
7.	Future Work	55
7.1	Immediate Interface Changes.....	55
7.2	ASR.....	55
7.3	Adapting to Mobility.....	56
8.	Conclusion	57

9. Appendix	58
9.1 A chronological list of accomplished milestones and features	58
9.1.1 Version 1	58
9.1.2 Version 2 (conceptual sketch)	58
9.1.3 Version 3 (sketch & partial code)	58
9.1.4 Version 4	59
9.1.5 Version 5	59
 10. Bibliography	 61

1. INTRODUCTION

Current mobile technology works well to connect individuals together at any time or place. However, general focus on one-to-one conversations has overlooked the potential of always-on group and community links. With the exception of IM and SMS, there is a void of designs that support group and community-level mobile communication. While IM and SMS are useful and ubiquitous, their designs are fundamentally limited by the minimal physical constraints of mobiles. The costs of text display and entry are too high to scale towards n-n communication sessions. Instead of a textual approach, I hypothesize that asynchronous persistent audio is a superior medium to support scalable always-on group communication. Furthermore, this medium is well suited for mobiles given their physical constraints. Audio is potentially a better medium than text for this reason. To evaluate this claim, one must first have an adequate interaction design before it is possible to investigate the qualities and usage patterns over the long-term. This design does not exist for mobile devices. This thesis takes the first step in this direction by creating and evaluating an initial design called RadioActive. If RadioActive proves to be a good design, we hope to prove with long-term studies the value of asynchronous persistent audio.

RadioActive is a technological and interaction design for persistent mobile audio chat spaces. RadioActive uses the notion of persistent chat space, which is a virtual place where individuals gather to exchange messages asynchronously. Mobility provides the ability to drop in and out of the conversations as the user pleases due to persistence.

Audio is well suited for sociable communication on mobile phones because 1) it is the primary medium that telephones are designed to support, 2) it is better able to convey emotion and personal identity than text, and 3) humans color speech with intention and degrees of confidence, enabling better inference of meaning and social processes (Chalfonte, 1991). Audio is more effective than text in group collaboration because of its ability to promote higher level and more social discourse (Chalfonte, 1991).

In RadioActive, it is possible to browse any message created since the inception of the chat space through a persistent archive. The archive permits out-of-order exchanges,

where one can reply to any message across time. This allows old conversations to re-emerge, with or without the involvement of the original participants.

RadioActive focuses on the key issue of navigating an audio chat space. In order to know what to listen to next, users need an impression of the chat space to guide them. Audio is slow and serial, making it difficult to traverse large volumes of messages. It is also much more difficult for machines to parse and interpret. RadioActive utilizes interactive visualization, where many of the many qualities of the chat space are presented visually rather than orally. A message and its subsequent replies, known as a thread, are represented by visuals that vary in size, shape, and color, to represent the underlying metadata. Users interact with these visuals using a touch screen to play and respond to messages.

Social uses of audio, such as the telephone, are typically synchronous. Research has made strides in scaling synchronous group communication (Rodenstein, 2002), but the medium's usefulness is constrained by the cognitive limits of concurrent presentation. Asynchronous communication is time-shifted, which instead allows all participants access to the cataloged information independently. Further, more of "the right people" can be active in the conversation by not constricting participants by who is available during a specific point in time.

In this thesis I examine related work, describe RadioActive from a methodologically constrained bottom-up approach, discuss the theoretical rationale behind the design, what seems to work, what doesn't, and suggestions for the future.

2. RELATED WORK

It is useful to examine and critique previous related work for design inspiration and to learn what to avoid.

2.1 NISHIMOTO AVM

The most similar system to RadioActive in both goal and medium is Nishimoto's Asynchronous Voice Meeting System (Nishimoto, 2002). AVM is a desktop-based asynchronous voice forum that uses Automatic Speech Recognition (now referred to as ASR) to transcribe messages. It uses a newsgroup-like interface where users can quickly browse the transcriptions. In case of machine error, users may choose to playback the original audio of user selected portions, much like ScanMail (Whittaker, 2003). However, they choose to implement interruptions, or "barge-ins", that provide an audio analog of in-lined quotations. They theorized it would reduce the total amount of communication required to discuss the same topic.

Their preliminary user study suggests that their system results in shorter and more frequent messages than a standard text-based forum when discussing the same topic. Users left 12 overlapping messages compared with 59 non-overlapping, suggesting some usefulness to the feature but not enough to make it a first priority in RadioActive. Analysis of their 5-point Likert scaling questionnaire pin points which approaches should be attempted or avoided. Users gave low ratings when asked if "the mood was very similar to when people actually congregate to conduct discussions", averaging 2.2 with AVM in comparison to 2.8 for text. Even with barge-ins that attempt to emulate face-to-face behavior, AVM was less like real-life than a textual equivalent! One explanation might come from the over-reliance on the transcripts compared to the original audio. Users reported they read more than they listened (4.0/5.0). This is not surprising given text is much quicker to read and follow on a desktop. Users also ranked the textual BBS higher in usability than AVM,

which likely has more to do with the interface design than the quality of the mediums.

Nishimoto tried to repurpose the design of email for voice, which is a poor approach given the differing properties of the underlying mediums. The mismatch is only intensified on a mobile screen. His navigational techniques rely on higher resolutions than are available.

Furthermore, converting audio into text strips away para-linguistic meaning, even if the ASR is error-free. Removal of any personal artifacts and expression further degrades the medium by detaching human connections. It is also doubtful that ASR could maintain low error rates when scaling in both message quantity and speaker independence, especially given background street noise. To avoid the shortcomings of AVM, this thesis does not use the concepts of barge-ins or ASR as a navigational or structuring agent.

2.2 SCANMAIL

ScanMail is a visual desktop-based interface to voicemail. Like AVM, ScanMail uses ASR to transcribe text for reading speed and to support search. It is useful for users with high volumes of voicemail to treat it like email in desktop environments. However, ScanMail does not provide any mechanisms for reply, a critical function of a complete communications environment.

ScanMail, like AVM, uses an interface design that is not appropriate for a mobile. We cannot easily display/manipulate large quantities of text or use similar 3-pane layouts while adhering to finger-sized interaction constraints. Both solutions mask audio's problems through transcription, which is inappropriate for large volumes of text.

2.3 MOBILE AUDIO CHAT APPROACHES

TattleTrail (Kim, 2002) and Symphony (Lakshmipathy, 2004) allow users to share a conversation space using mobile devices. Their goal is to enable small work groups to converse both synchronously and asynchronously in a linear sequential structure. A linear structure is a practical constraint on speech interfaces, as evidenced by the simplicity of the “catchup” function, which plays unheard messages chronologically. Sequential messaging structures also avoid the need for random-access browsing. However, the projects are relevant in their ability to move between asynchronous and synchronous. Such fluidity might be appropriate in future work.

Hyperspeech (Arons, 1991) and Hypervoice (Resnick, 1992) were systems designed to navigate highly structured hyperlinked audio data. They were the first research systems designed to demonstrate how hyperlinked content could be delivered using a telephone-based interface. The works focused on the navigation and manipulation of structured databases without a real computer. We can overcome their assumed constraints in telephone design with modern technology. Hyperlinking is useful concept regardless of constraints, and I will later discuss how RadioActive was augmented to use physical hyperlinks as a guide to a chat space.

2.4 BROWSING AUDIO

Audio navigational techniques often focus on expanding a sense of place in an audio-space. *Espaces2* (Sawhney, 1996) is an attempt to provide an audio-only environment using an “acoustic bubble” metaphor. Users browse hyper-linked audio data using 3D spatialized audio and audio icons. Spatialization and similar techniques such as earcons (Blattner, 1989), Sonic Browsing (Fernström, 2001), Ambient Soundscape (Kilander, 2002), and Audio Hallway (Schmandt, 1998)}, and parallel browsing (Arons, 1992, Fernström, 2001) attempt to multiplex or simultaneously present content in the audio domain. Such techniques incur large cognitive loads, which starve the other senses of attention. This could be dangerous in an urban or driving situation. Furthermore, it would be difficult to physically adapt these techniques for mobiles, given they often require precise speaker placement and require low background noise.

Although structural data has been conveyed using sonification techniques (Kilander, 2002, Mynatt, 1997), it is restricted in practice to one dimension of expression. Because the mappings are unnatural, adding any more dimensions loses meaning in perception. It is also difficult to separate or estimate data points given a large simultaneous spread in any number of dimensions. In addition to perception problems, the artificial mappings are difficult to learn.

All speech interfaces rely on the user to maintain state in working memory, which inevitably will be forgotten. Refreshing state can be as costly as the initial input. Even with the best simultaneous presentation algorithms, speech-based navigation is limited in the complexity and size of an audio space it can successfully convey. This can only be solved with a visual representation.

2.5 AUDIO SKIMMING

Time-compression techniques, such as Speech Skimmer (Arons, 1997), shorten listening time by applying DSP to a sound clip. The most common algorithms speed playback whilst maintaining pitch using techniques such as Speech Skimmer’s SOLA. Other skimming approaches remove extraneous parts of recordings such as silence and non-verbal utterances. Recently, Tucker and Whittaker used a more aggressive partial-removal tactic of linguistically processing ASR-derived transcriptions to remove large “irrelevant” portions of the audio recordings. They then applied strong time-compression (Tucker, 2006). This approach was justified by the goal of searching through hundreds of hours of recorded meetings. Their results were not favorable, especially placing very high cognitive load costs to understand the transcriptions. Even if it was understandable within normal cognitive loads for speech processing, the technique may not work for social environments where the messages are much shorter. Furthermore, in chat it is important to leave much of the message intact, as mutters and extraneous words function as social cues. To strike a balance between cognitive load and listening time, RadioActive post-processes messages to speed playback by 10%.

3. DESIGNING RADIOACTIVE

RadioActive exists to support on-demand social conversation amongst groups. As such, the related works are mostly inappropriate because their design goals are either significantly different or assume an incompatible set of constraints. It is more difficult to optimize the listening experience of many short audio snippets than long individual segments. Potential techniques are further limited by the low computational power and output abilities of mobiles. Expecting payoffs from the unique affordances of mobility, RadioActive’s principal strategy to support the medium is visualization. However, creating an effective visual interface for mobile devices is difficult. Visually representing hundreds of messages is not trivial on small screens, and mapping the underlying metadata to visual cues has few real-world metaphors to draw from. RadioActive attempts to solve these issues by using a clear compact representation of the audio chat space with a zoomable interface.

The visual abstraction represents the structure of the persistent chat space. Keeping navigational state in the visual domain has many strategic advantages over a speech interface approach. Speech interfaces require users to navigate blindly, projecting state transitorily and sequentially through the audio channel. Mentally retaining state is at odds with participating in the actual conversation. Instead, the visual representation retains the navigational state for the user.

Effective use of the visual domain also exploits additional computational abilities of users, as “spatial arrangements simplify choice, spatial arrangements simplify perception, and spatial dynamics simplify internal computation” (Kirsh, 1995). I have chosen not only to exploit spatial arrangements, but also other visual features like color that move useful affordance cues to the visual domain to simplify choice (Kirsh, 2004).

The visual representation conveys many dimensions of structure and metadata simultaneously to informatively guide the user through the chat space. Because we can simultaneously express more qualities of the space than possible in audio, users do not need to be restricted on a linear path like TattleTrail.

At the heart of the visualization is the unique representation of a group of related messages. Each message is represented as colorful circles (Figure 2). Variations in size, shape, and color represent the underlying properties of each message (Figure 1). These variations meaningfully differentiate posts through gestalt (Figure 3). When a user replies to an existing message, the new representative circle is placed near the parent post and connected with a gray line. This results in a graph-like representation for a group of related messages.

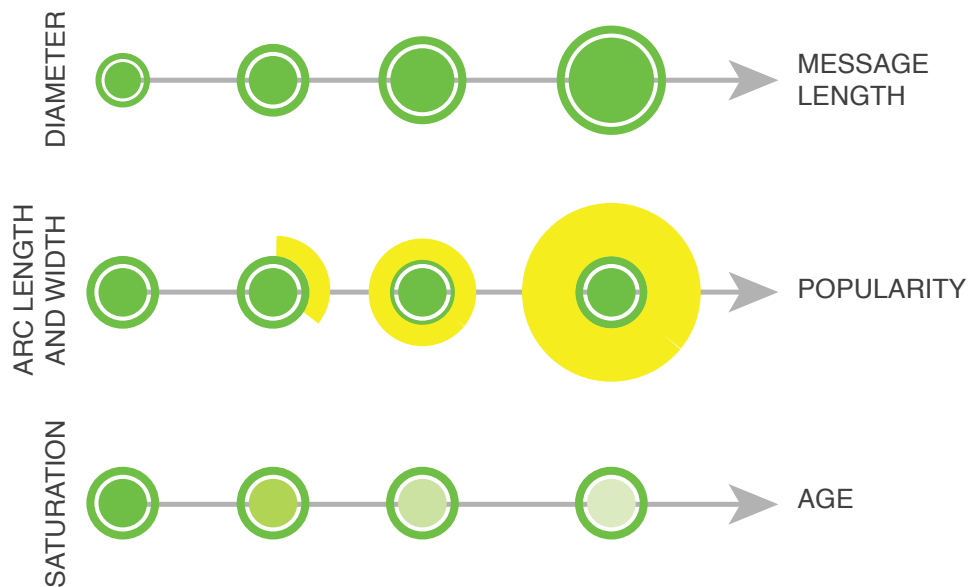


Figure 1. Circles visualize messages by mapping meta-data onto visual dimensions.

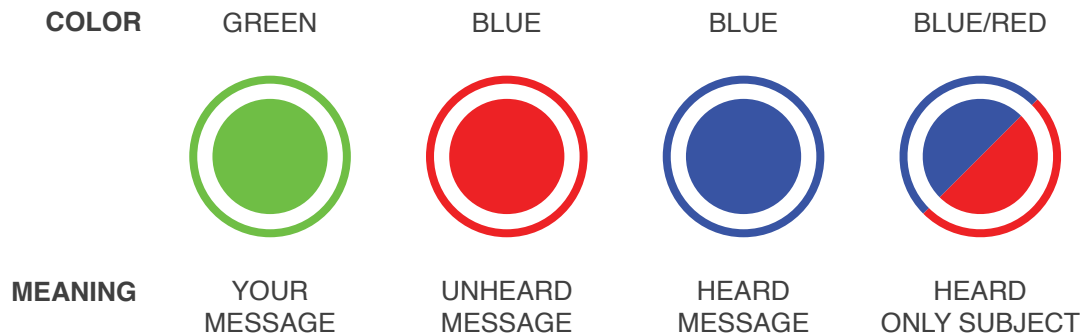


Figure 2. Changing the color effectively helps distinguish between messages and their status at a glance.

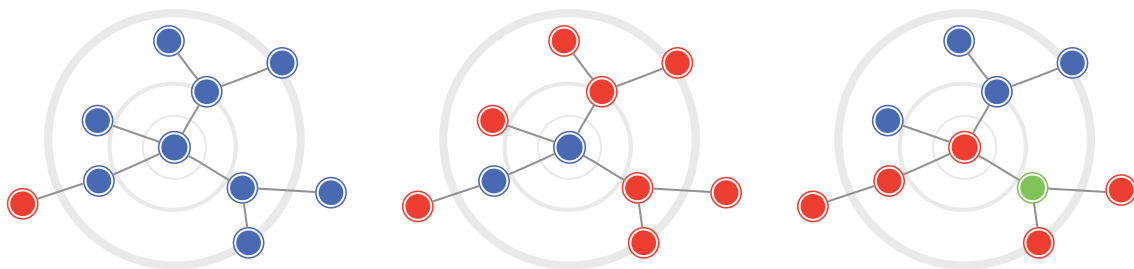


Figure 3. Three chat spaces with the same structure are visualized. Color effectively segments chat spaces by utility. Chat spaces with few unheard messages (left) are not as attention grabbing as a space with mostly unheard messages (middle). When only specific branches have been heard (right), color and layout-based clustering logically segments the space. Concentric rings help distinguish the first message, and function as navigational aid.

To address skimming, RadioActive uses structuring elements and humans to summarize posts rather than a machine-based approach. Longer messages are optionally accompanied by a short summarization, similar to subjects in email. Shorter messages only possess a body.

Users touch the circles to preview its subject or body, which is determined by the particular interaction style and the message's properties. After playback completes, a dynamic playlist is optionally constructed to automatically choose the next child. This lets the user put the device in their pocket and passively participate. The heuristic is informed by the user's listening history and the properties of the children. The resulting playlist is projected into the visual structure, available for manual manipulation (Figure 4).

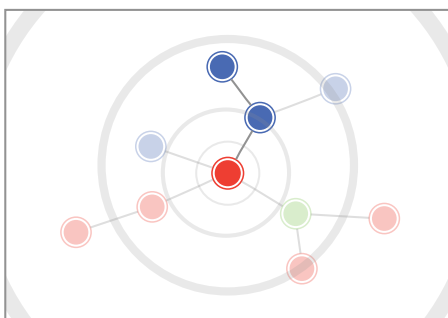


Figure 4. Because rings around messages and the lines are fully saturated normally, transparency works well to show and manipulate the current playlist.

3.1 DESIGN RATIONALE

I demonstrate the rationale behind the chosen design using a bottom-up approach by walking through the construction of a constrained chat space from scratch. The chat space is constrained by 1) the input and output characteristics of mobile devices, 2) not using ASR, 3) reliance on visual navigation, and 4) encouraging careful re-use of familiar interaction concepts. Detail is only added where necessary to clarify (Tufte, 1990).

3.1.1 CREATING A INTERFACE PALETTE

We start with an asynchronous chat space structurally modeled after Usenet, a text-based large-scale group chat system. We can't directly infer an interface due to the constraints of mobiles their usage contexts enforce: screen size, user's attention, ambient noise, etc. However, the concepts of how messages are related and browsed are adequately understood. In order to shrink the design, we can only maintain the most useful characteristics that act as navigational cues.

To decide which characteristics to preserve, we must first construct a gamut of potentially useful dimensions.

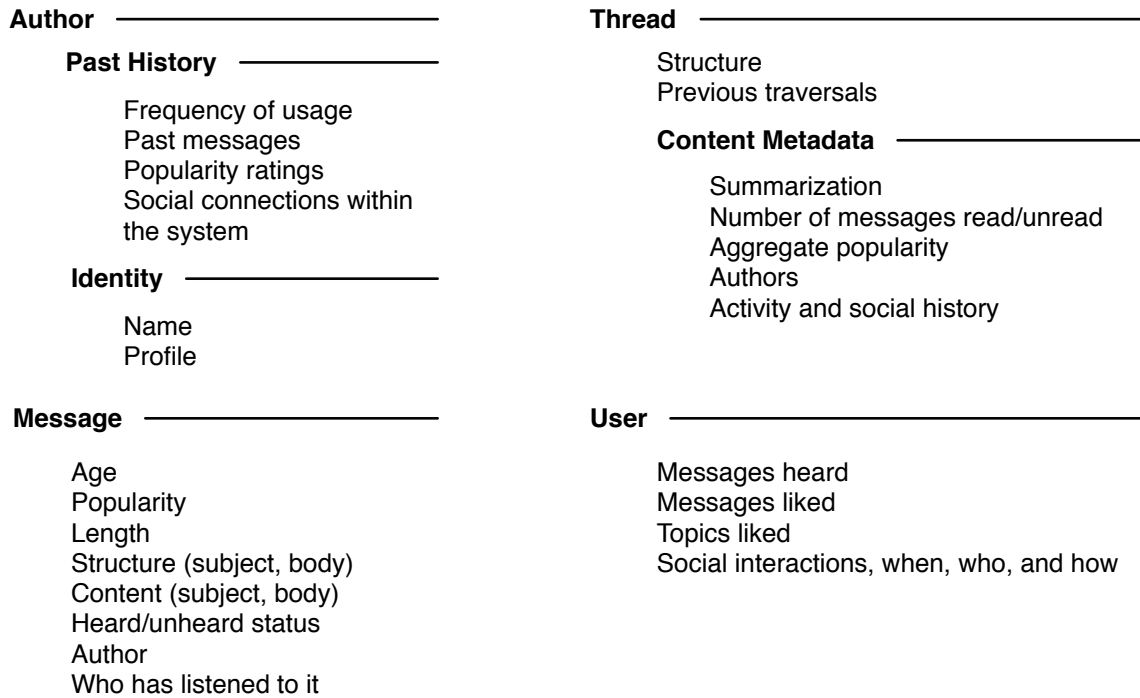


Figure 5. A first-pass list of potentially useful dimensions, including the most significant from email and newsgroups.

Now we have a palette of solutions to meet the design goals of each component in RadioActive. I will walk through in detail the visual construction of the navigational components along with their associated problems, starting with an initial invalid idea and its lessons.

3.1.2 THE CHAT SPACE

Creating the principals that guide the chat space, where the messages are presented, is the first problem that needs to be solved. The chosen representation directly affects navigation. The goal is to orchestrate a useful impression of a group of related messages, known as a thread. The impression enables the user to make an informed decision as to which messages are of interest. Using a touch-screen, the messages can be accessed with the user's finger.

The basic navigational principal later proved invalid was to spatially-ground the chat space (Viègas, 1990). Doing so concurrently shows multiple threads along a 2D plane.



Figure 6. The first version of RadioActive (left) used the spatially-grounded chat space concept from ChatCircles (right). ChatCircles was a synchronous chat space, where users could see messages in real-time provided they were within a spatially-mediated “listening range”.

How the threads are separated in space is up to the goals of the view. Because the goals of the user change, their arrangement needn’t be static: there are myriad heuristics for each problem. A reasonable start is to estimate first the principal component of the set heuristics that match user’s goals.

My estimate was to semantically separate space by topic. The exact heuristic employed would be up to the human. Observing how they use the space over the long term would emerge the best practices. One found they could then be automated. However, there were no best practices found in the evaluation because the user study participants disliked the entire concept.

Aside from being confusing at first, separating persistent messages in space by topic is actually a poor strategy when the set of topics is large. Humans lack the metaphors to usefully project a multi-dimensional ontology onto a 2D plane. The progression of ideas in space is fundamentally limited by the non-linear and multi-dimensional relationships of the real world. Clusters would have to eventually occlude their neighbors, impeding navigation. Furthermore, going left versus right only has meaning along the related arbitrary dimension for a specific group of messages.

While we’re deconstructing the original strategy, let’s also consider the use of infinite space. On a small screen, it is costly to traverse across threads with extraneous detail in between. We can solve this problem by dedicating each thread its own plane, called the

message-level view. A separate screen called the Inbox is used to navigate across threads.

3.2 GETTING TO THE CURRENT DESIGN

Let us start with the description of the message-level view.

3.2.1 MESSAGE-LEVEL VIEW

We describe a thread by its message structure. Previous designs provide a variety of starting points. The most common approach is found in modern email clients such as Outlook and Thunderbird.

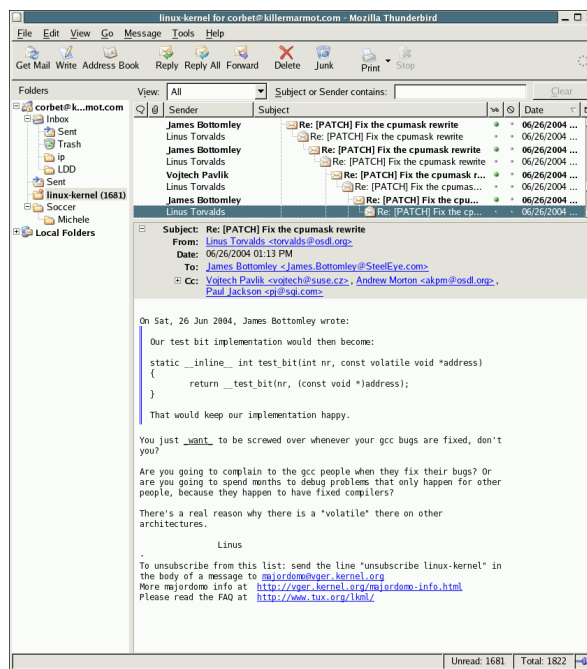


Figure 7. Thunderbird, like most modern email clients, clusters threads in a linear form. The layout of the interface and its priority on chronology reinforces a ‘reply-to-last’ behavior.

The layout is flat, 1.5-dimensional, and textual. This form is useful because often email threads are linear. Other methods include thread arcs (Kerr, 2003), Loom (Donath, 1999, and Venolia’s interface (Venolia, 2002).

Venolia’s method is a worthy starting point because the representation is compact and clear. However, we must consider how her representation works on small screens and a different medium. Her design prioritizes chronology, always appending new messages down and to the right.

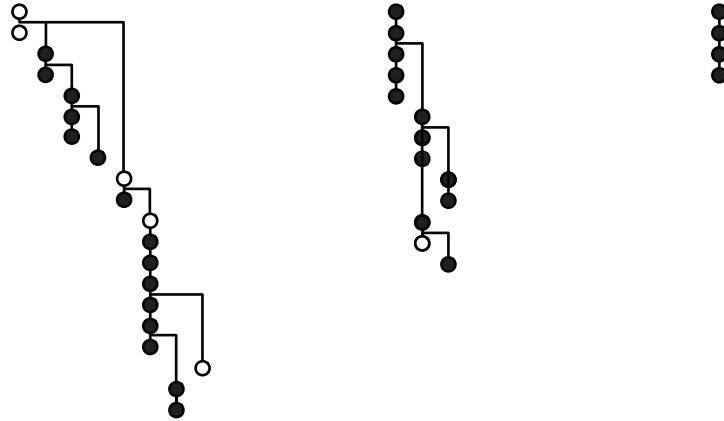


Figure 8. Venolia’s thread visualization gives priority to chronology, a quality important in email. Each message sits on its own row, while branches share columns provided its contents do not overlap with the branches below it. The non-interactive visualization functions is to provide a brief visual summary of branching, quantity, and read/unread status. Due to the vertical bias and de-emphasis of clustering sub-dialogues, it is inappropriate for persistent mobile conversations.

In the scenario where an existing thread has a new reply to the root message, this layout will result in a large vertical branches. The more multi-threaded the messages become, the more difficult it would be to follow.

This layout also reinforces the “reply to last” usage pattern found in email. Any other behavior leads to complex branching. Given the speed of audio, I felt it was important that the interface biases the user towards leaving short messages more frequently. This approach uses humans to segment the conversation by context.

3.2.2 ENERGY-MINIMIZING LAYOUT

Ignoring any user behavior biases in her layout, Venolia’s method is inappropriate for RadioActive due to the small screen constraint. Her layout grows in one direction rather than trying to fill all of available canvas. We can solve this problem by using an energy-

minimizing layout, keeping her circles and lines. This layout minimizes the distance between each message using all directions to better fill the canvas.

The result has several advantages: 1) it emphasizing clustering, 2) it minimizes the display size constraint, and 3) the resulting layout has a more organic aesthetic. We further optimize the display by biasing the layout to fit within a specific width:height ratio. By matching the ratio of the actual screen to the layout, the complete graph will exactly fit when complete zoomed out.

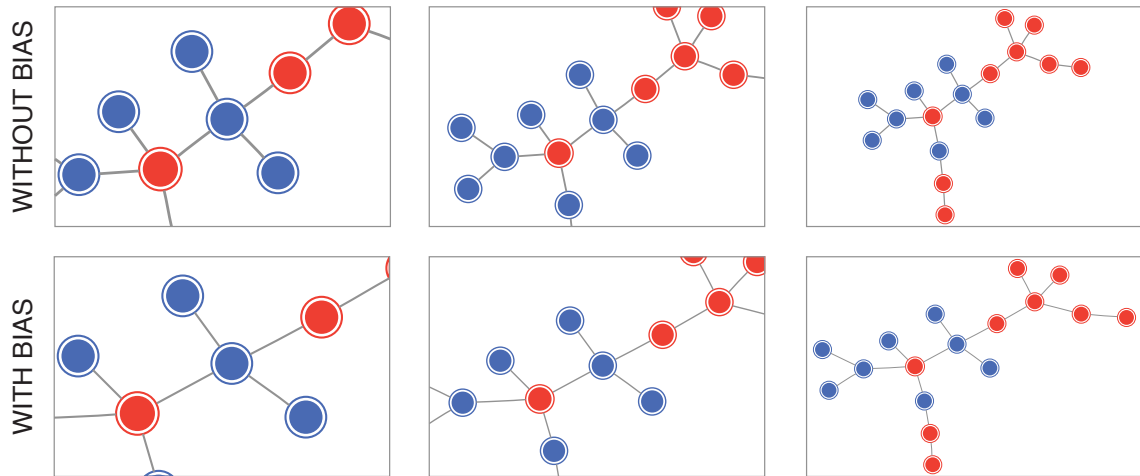


Figure 9. A visualization of a multi-threaded conversation using energy-minimizing layouts. Without a width:height bias (top), message clusters are tighter and do not take the screen size into account. Using the bias (bottom), the layout will always exactly fit on the mobile screen when fully zoomed out.

Energy-minimizing layouts also help navigation. Spatial distance from the root post gives an easy approximation of reply-level depth, something that is difficult to estimate with Venolia’s method. Further, by spreading possible routes away from each other we simplify choice: unrelated messages are less likely to cluster.

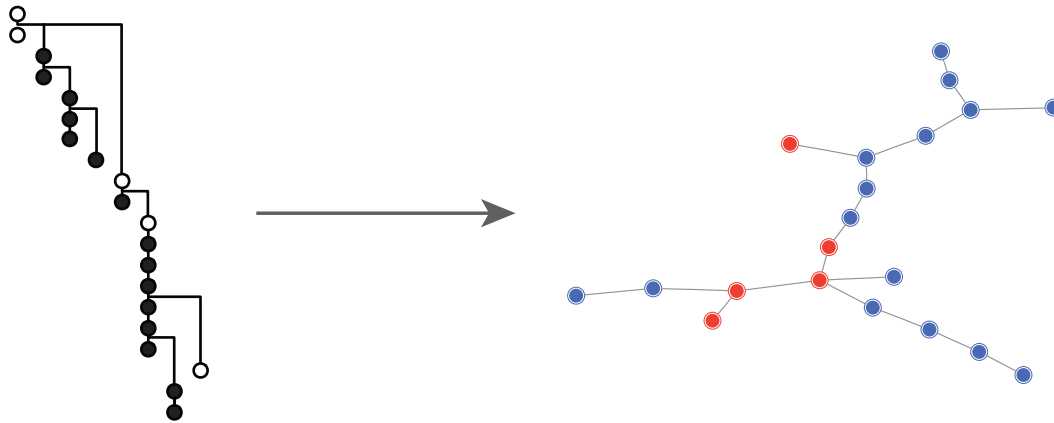


Figure 10. Venolia's layout (left) works well to show linear threads, but does not cluster non-linear threads as effectively as RadioActive's energy-minimizing graph (right).

3.2.3 COMMUNICATING MESSAGE HIERARCHY

Venolia's method transparently illustrates message hierarchy because children are always placed downward. RadioActive's chosen graph-like representation is presented as if undirected. Several attempts were made visualizing message hierarchy (Figure 10), however it was felt they made the entire visualization too busy or was too subtle, losing the valued aesthetic. The indicators can be so numerous they visually function as noise.



Figure 11. Three different sketches showing the original attempts at communicating message hierarchy with a thread.

The chosen compromise was to stroke concentric circles of increasing distance originating at the root post (Figure 12). The circles function as a guide, where the tangent of a curve is orthogonal to the root post. The radius gives an estimate how far it is. Given the limited screen size, this approach matches many of the goals of a representative map in the corner, as was done in the first iteration of RadioActive (Figure 6).

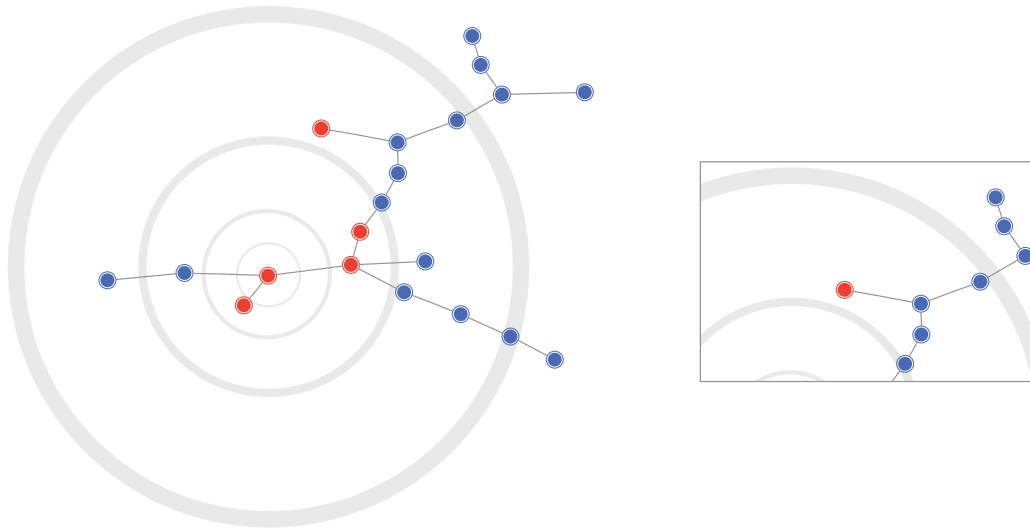


Figure 12. Concentric circles simultaneously isolates the root post (left) and serves as a guide within the displayed portion of an infinite plane (right).

3.3 MESSAGES

Now that we know how to layout and connect the structure, the main unknown remaining is the message representation. Like threads, there are many qualities on our palette, in addition mappings to visual dimensions.

To guide the design, it is worthwhile to first examine what might be presented in audio that could be shown in the visualization. Audio is slow, so it is important to know if the payoff of listening is worth the time required. Worthiness is a function of message length and quality of content. A representation for each would allow the user to estimate worthiness without having to listen to it.

3.3.1 REPRESENTING AUDIO: MESSAGE LENGTH

To depict message length, we could use either textual description, such as “1:28”, or vary a visual dimension like length. Text descriptions require sequential reading, and therefore do not effectively present a global impression. Instead, I first conceived that a thick line was a reasonable way to represent a message. To represent recorded length, the line length would be proportionate against some scale.

The line easily is augmented into a timeline because the visual form supports sequential data. For example, the timeline could be used for Nishimoto’s barge-ins, showing where a reply was initiated (Figure 13). But given our constraints of screen size and avoidance of barge-ins, the message representation has to be more compact than the lines.

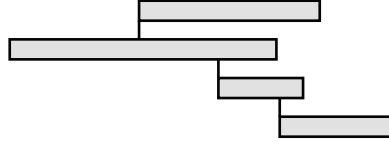


Figure 13. Lines work well as a message representation to communicate message length. Lines are also easily overloaded to function as a timeline, showing the point of context for a reply.

This reduction can be achieved by mapping message length to area of a circle, keeping the said part of Venolia’s method. To compensate for additional area generated when X and Y dimensions are simultaneously increased, squaring perceived area, we use a non-linearly compute the radius:

$$width = minimumSize + \alpha \sqrt{\frac{lengthInSeconds}{secondsPerInch}}$$

where 1 unit = 1 inch, minimumSize ensures the circle is visible, lengthInSeconds refers to the message, and the constant secondsPerInch controls growth at rate α .

Changing these constants affects how the circles grow within the chat space. They must be appropriate for the screen size in terms of both ability to discriminate with a finger and area taken from other messages. Such constraints can be calculated and imposed by the designer, but the decision must be informed by the resulting bias it places on user behavior.

In email, long messages do not occlude other messages. Message length is irrelevant to a message-level view. In RadioActive, messages that are longer or shorter than the norm will stick out and occupy valuable real estate. This visual behavior functions as feedback to the quality of the message length, based upon the judgments of the human participants.

This is illustrated by a simple example. Suppose any messages over one minute are huge, and everything under a minute is of perceptually normal size. Over enough time, the

difficulties imposed by users that create huge messages will adapt to accepted practices of not going beyond a certain threshold.

To calculate the initial seeding values, it is most important to first think about the ultimate constraint: finger size. If you can't reliably touch a circle because it is too small, the interface is useless. A conservative minimum for the user and a touch-screen to discriminate between objects is 0.5 inches. With an absolute minimum, we can calculate our area growth rate to favor longer messages. Let us assume the desired average length is two minutes, with a radius of 2x the minimum, or 1 inch. Because we don't have enough usage to know how we want to influence message length, we leave α alone by setting it to 1.0. We can then solve for secondsPerInch, which is 480. The final calculated width in pixels is obtained by multiplying by the DPI of the screen.

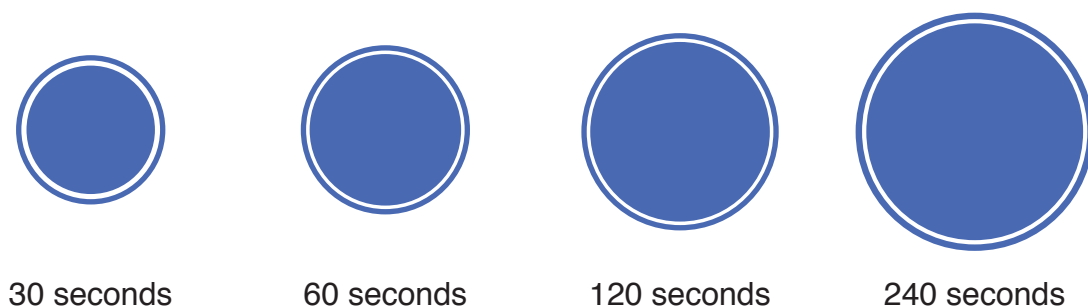


Figure 14. Example messages, varied by length, in their actual size as shown on the screen.

3.3.2 REPRESENTING AUDIO: MESSAGE CONTENT

Now that we have a representation for message length, it would be useful to visualize the content to make it easier to skim. However, the only passive way to extract it is via ASR. Because ASR is not within the scope of this thesis, we must leave it for future work and instead be more creative in our compromises. For example, we could ask users to input a summary of each in text. Unfortunately text input is a constraint, but we can ask users to summarize in voice. Because message creation is forcibly linear in audio, unlike email, we ask users to summarize what they have just previously said. While previewing voice is not as cheap as text, the short segment provides a gisting mechanism, encouraging exploration.

Another way of assessing message value is popularity. Unlike text, by streaming the sequential message what percent is globally heard. This is used to derive popularity with-

out requiring any extra effort from the user, unlike Slashdot. We calculate a value range where most messages fall between 0-100, but they can go higher. Popularity is calculated with linear weighting according to the following formula:

$$popularity = \frac{numListensForThisPost}{20 \cdot globalAvgListensPerPost} + (70 \cdot avgHeardPercentForThisPost)$$

We visualize popularity with a partial ring around a post, increasing its thickness where popularity exceeds 100 (Figure 1).

3.3.3 OTHER ATTRIBUTES: HEARD/UNHEARD STATUS

Message length and knowledge of its content are the two message properties stored in audio. There are plenty of meta-data that the system is aware of, such as which messages are unheard. We can give a better impression by adding more detail.

Representing which messages the user has heard is an essential message attribute. Email clients typically show and hide a bold dot (Figure 8). However, this boolean visual descriptor cannot describe all the states in RadioActive. Relative to a particular user, each message is 1) completely unheard, 2) partially heard (subject versus message), 3) completely heard, 4) message was authored by the user.

Instead of black or white, we can use other colors to differentiate between the states with gestalt. Color is a good choice because not only does it give us a nice look, each the categories are well differentiated using primary colors. We split the four states across three main colors such that they are reasonably separated along the color wheel to help gestalt: blue for unheard, red for heard, and green where the user is the author. The colors for “heard” and “user is the author”, blue and green, are closer together on the color wheel than red to help give family effects. Red metaphorically grabs attention.

Colors work well because they can be broken down into more dimensions. We’ve mapped message status onto hue, which leaves saturation and value. To indicate message age, we can use saturation with another metaphor: older messages fade out. However, humans can only coarsely judge de-saturation. This is not as much a problem because RadioActive is a social system. Thus, it is not important to have high-resolution timestamps when only a rough relative judgment is needed. We can still improve age estimation by giving a reference point. Encasing the message with a fully saturated band yields two visual effects: 1) estimating the level of de-saturation is aided by an extreme, and 2) it helps read/unread status further pop out. The second effect is important since the perception of hue differ-

ences weakens with decreasing saturation (Figure 1).

3.3.4 OTHER ATTRIBUTES: AUTHORSHIP

It is always helpful to know the author of a message. This request specifically came up in both user studies which lacked it in the representation. However, authorship cannot be mapped onto a range of 1-dimensional values like age or length. Original designs left out the author's name from the visualization, relying on stating the name in audio before each post. After the second user study, names in text were put directly in the middle of the message to help people scan for a particular author. While this increases the amount of detail and bounding box of the circle, it is permissible because names have an upper bound on length.

3.3.5 MESSAGE INTERACTION

Now that we have a representation for a thread and its messages, we need to design our interaction strategy. Arrow keys are too limiting to select messages with random-access in a visual space. Instead, we rely on touch-screens, available on select mobile devices. It is still unclear what style of interface or combination will prevail in the future, so it is appropriate to assess the potential of the richer input source. Touch-screens are often disliked because the displayed elements to choose from are too small. Enforcing thumb-based minimum sizes for objects throughout the system, we enable users to whip out their phones and instantly manipulate a chat space without the awkward stylus. The chosen reference implementation platform, the Motorola A1000, lays out its UI in portrait-mode. However, the physical design makes it awkward to single handedly use ones fingers in portrait. By switching to landscape, we naturally encourage two-handed usage where the thumbs can access the entire screen.

The obvious interaction strategy is to press messages to play them. However, we can extend this a bit further to deal with the separation of message and subject. When a user taps on a single circle, it plays the entire message, skipping the subject if one exists. If a user holds onto a circle for at least 300ms or starts to drag without letting go onto another circle, it plays the subject of the last circle hit. If the message lacks a subject, its body is played instead. To identify which message is playing, we animate small rings around the associated post (Figure 15).

Starting a drag in empty space results in translation, much like Google Maps. This helps deal with the limited screen real estate on a touch-screen capable device. The arrow keys also map to translation of space. The other method is zooming (Bederson, 1994), sup-

ported by buttons on the side of the device as well as in the interface (Figure 16).

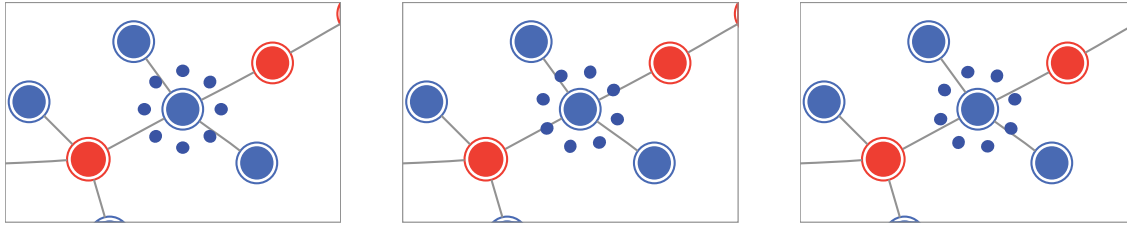


Figure 15. Small rotating rings appear around the post currently playing.

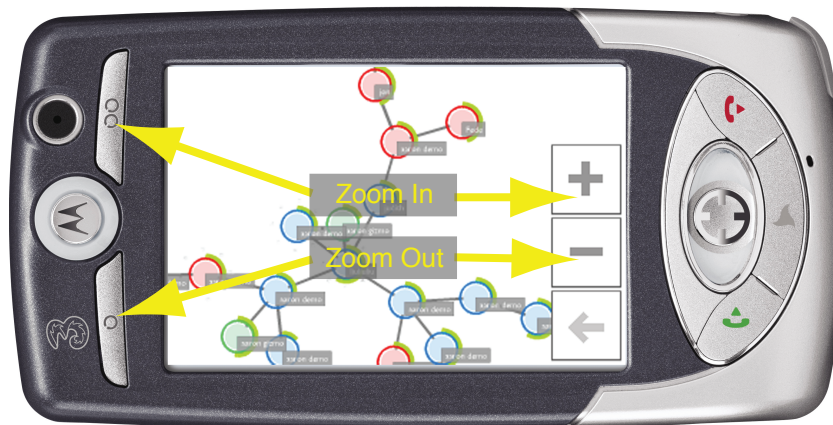


Figure 16. Zooming is controlled by buttons in software and hardware interfaces.



Figure 17. From left to right, the sequence of replying to a message. The user initiates their reply to the message currently playing. A voice message then tells the user to “record their message after the tone”, followed by a confirmation screen after the user has completed their recording.

Users reply by clicking on the appropriate button during playback of the parent message. Users are then prompted to record their message, pressing stop when they are finished. They are then prompted to keep the recording, cancel, or redo it (Figure 17). The message is also played back during the prompt. If they choose to keep it, the layout is re-calculated, and the view centered on the new message. Message playback begins again after the centering to help verify the new message is in the correct place.

3.4 INBOX: COMPACT THREAD REPRESENTATION

Now that we have a message-level view to browse the contained messages, we must look for interfaces that summarize threads. Typical email clients automatically cluster messages by lineage, assuming the oldest subject throughout the whole conversation. In RadioActive, we ask users to voluntarily input a series of related tags in text during the creation of a thread. This single place of text entry in RadioActive is worth its cost to help users find a thread of interest. The text is the fastest way to skim groups of messages.

Tags and folksonomies use humans to segment the topic space. Humans, error-prone themselves, are still the gold standard at classification. We can reasonably use tags to navigate content on mobile devices. The words themselves use few characters, and are already in a machine-parseable format. Visual filtering can help trim down the data sets across many criteria. To limit scope, this thesis does not fully explore tag guidelines and opportunities for mobile phones and conversations. This remains future work.

Instead, we aggregate all of the threads into a discussion space (Figure 18). Discussion spaces can be constructed or constrained from the set of all messages by a number of factors. Part of the power of RadioActive is its general design that can adhere to new contexts. The eLens project, which will be later discussed, used physical locations and social networks as URLs for a given set of threads, which also encompass the notion of a chat space. The threads can also come from more typical sources, such as through a higher-level ontology of topics like Usenet.

But if we are more dynamic about the construction of an inbox, we could better augment mobility by subscribing to many different physical contexts. Thus the relationship of a thread to a chat space is fuzzier, leading to multiple instantiations of a message or thread. This has subtle side effect. Imagine a chat space that exists for anyone within 5 miles. This rule constraints who can participate and who cannot. With respect to privacy concerns, lets assume Alice is within the requisite 5 miles, and she hates Bob who lives 10 miles away. Because of the way she entered the chat space, Alice thinks that Bob cannot access her messages. However, if some other search term or fuzzy context yielded a

ten-mile radius for Bob, Alice’s assumption is broken. To avoid complications, we assume each chat space to be unique, messages belong to one thread, and threads belong to one chat space.

Given our constraints and tag-based summarization, we continue the design with a minimalist perspective. The only additional meta-data provided with the tags are read/total counts and averaged popularity. The message counts are displayed to the right of the tags, with a green bar in the background to represent popularity. This set of dimensions provides information on a thread’s usage, and enough some information estimate the content. The threads are stacked line-by-line on top of each other, sorted by most recent posting. When a thread is selected, the separate vocal subject of the root message is played. This screen is therefore accessible with only audio and a directional keypad.

During the primary user study, we had in place an alternative thread screen that used a graphical representation to show the proportions of read/unread messages in a thread at a glance (Figure 18). It was created under the assumption that users highly valued threads that had new content. The textual summarization would appear only after clicking on a thread’s representation. However, the other view is also sorted by last updated thread, so the user is always able to access the newest messages without obscuring indications of content.

In a user study, the alternative thread representation was also found to be somewhat confusing, and informed the decision to make popularity and topic summary the soul priorities of visual presentation. While the text summaries require additional time to read them, the on-the-go scanning that was envisioned for the colored thread boxes can be better served by looking at popularity graphs that are useful and easy to read. Future work might examine supplementing popularity values with a recommendation system for even more relevance.

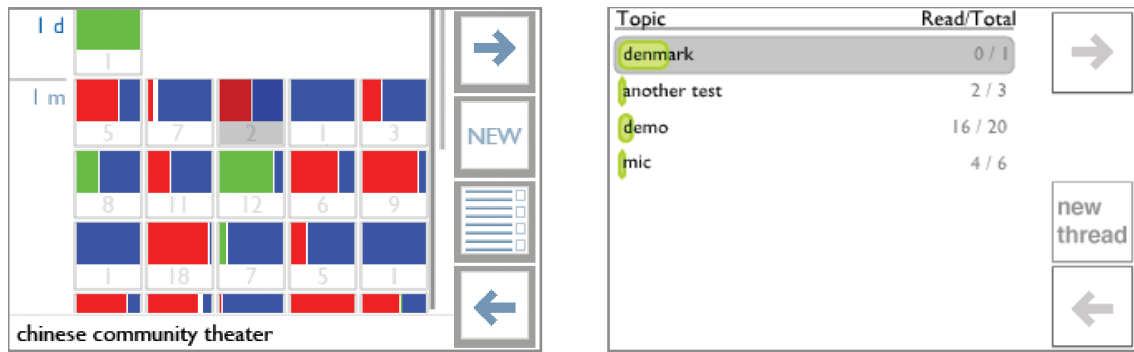


Figure 18. The original inbox (left) simultaneously shows many threads, providing a quick graphic feedback on the user's history within the space. User studies revealed it was more opaque in usability than the redesigned interface (right).

3.5 OTHER DESIGN ELEMENTS

When designing a non-linear and high informational volume environment, it is important to only present the best set of actions and goals perceived possible. Such a restriction is more significant when the cost of input and output is high. RadioActive's design tries to meet this goal through clear visual design and navigational techniques to maximize the human's intelligence as a guide.

RadioActive separates actions and objects by distinct visual family differences. Buttons are always on the same side using a similar graphical motif. Locations of common buttons such as go back or forward are consistent in location to match user expectation. Icons were designed to make the intention of a feature obvious. Initially, actions were separated completely from objects. Users had to click an object and then another button to perform the action. This was to make sure big thumbs on a small screen would do the right thing. In the eLens evaluation we learned this approach was unintuitive for users, who expected to be able to manipulate the objects directly. It is now the case that all objects have inert default actions assigned to them, in addition to their actions on the side.

3.6 NAVIGATING WITHOUT TRYING

Sometimes it is useful for typically interactive systems to use some basic intelligence to free the user from always engaging and selecting the next action. After using the visual

interface to setup a context, RadioActive provides the ability to let system take control and function like interactive talk-radio, hence the name RadioActive. Based upon simple structural features, we can generate a playlist from a starting state to continue the playing the discussions (Figure 4).

The algorithms necessary to seek the most interesting messages can be infinitely complex. More tangible features, such as if a message has been listened to, its popularity value, its length, its author in comparison to our favorites, and number of replies, are within easy reach. However, we can start planning paths according to the attributes of grandchildren as well, including the grandchildren's children. More advanced algorithms (requiring extra infrastructure) might use topic spotting and discourse analysis to yield context for comparison against an interest profile. At an extreme, we could even analyze messages for affect, choosing messages that fit best against a user's psychological profile.

Thus, there are many ways we can think of trying to figure out what people want a priori, but ultimately this is something that cannot be known or predicted by the computer. Therefore, the developed algorithm seeks to maximize both utility and simplicity. It functions greedily by ranking the children of the current node by popularity and choosing the highest, with some exceptions. The popularity of a child is examined only if it is unread. If the post has already been read, it can only be chosen if it has a child that is both unread and is popular. Quality is manipulated with a variable minimum threshold, chosen by the user. This algorithm has been chosen because it is easily comprehensible by the user, gives a linear presentation, and attempts to give an interesting and small sample of a thread.

3.6.1 TRANSPARENCY IN MACHINE-INTERFACES

When machines have “too much intelligence” or too complicated a design, it can be very confusing to the user to understand the state of the machine and how it got there. It is a design choice in RadioActive to make automated playback as simple and clear as possible. It is useful to use an algorithm that is both obvious to the user and warranted. Thus, playlists should be first improved through wise selection of the influential dimensions. If the algorithm uses too many dimensions, this process is opaque to the user, making the interface harder to use and manipulate. Assuming the system is well used, fully browsed, and unbiased, I hypothesize popularity to be the most useful and clear metric. Further work is needed to evaluate this claim.

3.6.2 LINEAR PRESENTATION

I have already presupposed the next post chosen is a child of the current, yielding a linear presentation. It is the intended usage that that messages are posted in a context inherited from its ancestors. If the presentation skipped messages, the missing context can be jarring and confusing even within a single linear depth-traversing path. The user always has the option of manual random-access and non-linear presentation by escaping the radio mode and navigating only with the visual interface.

3.6.3 CHOICE OF COVERAGE

We can take either a more breadth-first or depth-first approach towards playing back message trees. The answer is bound by the goals of the user: does the user want a glimpse of all response chains to a message, or are they more interested in skimming across entire threads? In this project, I've opted for the later approach to give a limited impression of what is interesting, avoiding breadth-first because the user can use the visual interface to further drill down at any point. A breadth-first or more interactive approach is best with a speech interface approach to the entire conversation space.

4. DESIGN THEORY

With a design in hand, I will now examine how it fits within conceptual frameworks of human behavior and navigational strategy.

4.1 INFORMATION NAVIGATION

In “Running Out of Space: Models of Information Navigation” (Dourish, 1994), Dourish suggests there exists at least three navigational domains to manage large bodies of information: social, semantic, and spatial.

4.1.1 *SOCIAL*

Dourish defines social navigation as occurring when “movement from one item to another is provoked as an artifact of the activity of another or a group of others”. Examples include exchange by word of mouth and collaborative filtering. RadioActive uses social navigation by making the history and activity of past users more transparent by exposing popularity. Future work might examine other methods of clustering such as by author or recommendation to further enhance navigation in RadioActive.

4.1.2 *SEMANTIC*

In general semantic navigation is accomplished with user-provided or computed metadata, and through hyper-linking the results of feature extraction. Features can be extracted through natural language processing or other discourse analysis techniques, such as summarization and prosodic feature analysis. This type of navigation was purposefully avoided to both limit the scope (these topics are PhD worthy on their own) and to understand what can be done without extra erroneous computer-based analysis.

Using ASR or any other machine-based subject analysis cannot be plugged in directly: the small screen of mobiles limits concurrent text display. We don't always need better machine-learning or visualization techniques to deal with these circumstances if we can rely on human intelligence for summarization. RadioActive asks users to summarize their messages and threads by voice and text, respectively.

A systems-level view lets us see that these costs will be incorporated into emergent standard practices by the community because 1) voice takes long to listen to, 2) typing is costly, and 3) text uses a lot of display space. Such practices are likely to be grounded on the expectation that users summarize usefully. We further support this by acquiring summarizations after the user has posted and confirmed their message. At that point, users know what they said and have it in working memory.

The computer scientist would look to Ben Shneiderman-style approaches or AI for semantic navigation. It is important to remember that human behavior can be manipulated (or more precisely, the constraints and affordances of their activity space), and thus semantic navigation should be additionally designed from a systems-behaviorist perspective.

4.1.3 SPATIAL

Dourish differentiates between information that is “inherently spatial” and mapping a semantic relationship onto a spatial arrangement. When looking at the data independent from any geological sources or urban applications, RadioActive lays out raw messages spatially to represent their semantic relationship: the root message and its replies. Dourish specifically encourages examining the appropriateness in a design, as spatial mappings are limited by

[...] dimensions—geometric, absolute, orthogonal. We must either restrict our choice of information dimensions to those which share those properties, or build a system in which spatial discontinuities or inconsistencies will arise. This is problematic when semantic and social navigation are seen only to take place as a result of spatial organisation. In realising where navigation is actually semantic or social in origin, we can avoid geometrically-based constraints to which spatial models are subject.

In conversations, the parent-child relationship between messages is not constrained by spatial dimensions. We have room for more information overloading as not all visual dimensions are dedicated to node relationships. Future work might look at more ways to

extend spatial navigation by examining the potential of physical space as a data filter.

4.2 SPATIAL DYNAMICS

Given we are able to navigate in a way that is consistent with Dourish's recommendations, we also should note how our choices reflect spatial abilities of humans.

We know that “spatial arrangements simplify choice & perception” (Kirsh, 1995). Using an energy-minimizing graph, we can help spatially cluster related messages to provide easier indexing ability within the users mental model. Additional structure can be projected through gestalt. While not strictly arrangement, descriptive small sets of color helps organize and arrange the affordance landscape. For example, a branch of unread messages from an already heard popular message forms a useful visual cluster.

Spatial dynamics help people perform computation on objects more quickly than in their head for set manipulation tasks (Kirsh, 1995). This could be appropriate in manipulating playlists to help people sort interesting messages from the uninteresting. This approach is worth examining more in future work. The scope of this thesis attempted to only use gestalt principals to organize cue structures as much as possible. This restriction helps us understand the limits of a simpler interface to address usability concerns. Future work might examine allowing users to interactively drag, delete, sort, expand, and manipulate threads, messages, and topic clusters, as well as using topic spotting to auto-collapse unrelated portions of a thread.

4.3 CONSEQUENCES OF AUDIO

The system primarily assumed voice as a way around an input/output constraint. However, we should consider the other effectiveness of voice on communication.

Voice often feels like it provides a better insight into the mental state of the author, even if it actually doesn't. While an audio system might permit users to delete and re-record messages as often as they wish, it is much easier to seamlessly change a message with text because of non-linear editing that doesn't interrupt flow. Due to the forced repetition of an entire message, or the cumbersome actuality of sentence deletion, it is more likely active users will send less calculated messages. Even in the unlikely event that a participant writes out the message before reading it into the interface, the inability to encode prosody in text means that the author will have to rehearse the message to achieve the idealized

perception. Thus the cost of creating the perfect message in audio is far higher than text. As a participant in the chat space, this can lead to feeling more connected, or at least a more honest discussion. Connell et al. notes, “two important aspects of communication media affect impression management: (1) with fewer incoming social cues, public self-awareness and the tendency to regulate one’s own behavior are reduced, and (2) with fewer outgoing social cues, a person’s ability to control how his/her behavior comes across to others is reduced” (Connell, 2001).

The flip side is that now people on the receiving side are aware of this phenomenon, so how they read and comprehend the message might be tainted by additional suspicion or a false interpretation of the author’s mental state. Thus audio can increase the probability of miscommunication even though it provides more meta-communication than text.

5. TECHNOLOGICAL DESIGN

Now that the interaction design has been discussed and rationalized, I will describe the technological framework that powers RadioActive. It is structured by a client-server model.

5.1 CLIENT

Choosing a platform was not easy for RadioActive. The state of tools available for developing large systems on mobiles is poor. The problem isn't as much the slow speed of the chips (which continually improves), it is the lack of integration and adequate software stacks. The only reasonable option for portable code is to use Java, which is on nearly every cellular phone. However, the actual implementation differs as hardware vendors often write their own alternatives to JSR standards.

It was decided for the eLens project, described in the evaluation, to use Motorola A1000s. They had a faster processor (200MHZ), touch screen, speakerphone, 3G speed, camera, GPS, and Bluetooth. The 3G was important because allowed for dual-voice and data, such that all audio could be streamed across a normal phone call optimized for low-latency. Cellular data channels suffer from random network jitter because the network treats it with a lower priority.

It was necessary to write a C-based program that locally talked over TCP to interface with the GPS, camera, speakerphone, and application focus. However, most of the RadioActive could be implemented in Java. Java allows for quicker prototyping than Symbian-based C, portability across devices including web and desktop, garbage collection, and the ability to easily plug-in 3rd party libraries where possible for additional functionality.

J2ME, or more specifically CLDC 1.0 and MIDP 2.0, is the target Java environment for the A1000. This set of libraries is not helpful in providing support for more advanced applications. The main limitations are 1) a lack of floating point, 2) lack of a GUI, and

3) lack of a proper 2D drawing API. Separately, the A1000's virtual machine lacks a JIT-engine and a generational garbage collector. This must be dealt with by limiting function/stack depth and object creation.

In order to work around many of these issues, I co-designed a framework called Barcelona with Sajid Sadi. Barcelona is a vector-based GUI toolkit designed for mobile applications running on smartphones. It is written in pure J2ME CLDC/1.0 on top of TinyLine 2D: a 3rd party library designed for SVG content on mobile phones. TinyLine's licensing permits use in both non-commercial & commercial contexts, and is thus appropriate to create a framework on top of it in the interests of both research and lab sponsors. Because TinyLine itself uses almost no Java or J2ME classes, it is suitable for both desktop and mobile development. This speeds development time considerably.

Graphics are rendered to a pixel buffer, which is in turn copied onto the display. Barcelona uses its own abstraction of screens and widgets, using a common TinyLine renderer to generate both GUI elements and custom application-specific graphics. Because the rendering model is unified with the toolkit, it is easy to extend existing components with custom canvases. J2ME's native graphics routines don't even permit setting the line length, let alone alpha-blending or Bezier paths. Using TinyLine, Barcelona can use alpha-blended Bezier-paths and anti-aliased text. This freedom is extremely helpful to bring novel designs to mobiles.

The basic Barcelona GUI toolkit provides customizable lists, buttons, labels, on-screen keyboards, text fields, scrollable areas, clippable areas, and menubars. A minimal API exists to help layout, but in the interest of time a more complete layout system was not design. Layout is mostly unnecessary because the size of the display is static.

All visible widgets and regions inherit from class UIElement. UIElements are attached in a graph to a Screen, which is in turn rendered through a GraphicsAdapter. The GraphicsAdapter abstraction allows us to easily target different types of displays, such as web, phone, or desktop application. Furthermore, because the widgets are unified with the accessible scene graph, advanced features like geometric transforms can be applied to the UIElements themselves. This design made it trivial to implement zooming of the chat space.

Barcelona had to be designed with the phone's limitations in mind. In particular, it was necessary to require fixed-point math universally, as the VM does not support floating point. Another design constraint is the lack of real multi-threading. This obstacle is dealt with by minimizing the number of concurrent threads. Inspired by Java's Swing, a main event loop serializes all operations. As suggested by Sun for mobile development, the

main thread is free to process display and input events. Events are cached in an event pool and appended to the main event queue.

As J2ME does not provide non-blocking I/O, separate threads for reading and writing to the network run alongside the main thread. This network handling method was found to be efficient, consistently able to reach bandwidths of 30K/sec with cross-Atlantic pings averaging 300msec.

5.2 SERVER

The RadioActive server coordinates a sessions, defined by a junction of data and voice channels. Sessions are initiated by the client to the server. All data connections must be initiated by the phone itself, as mobile networks conceal internal IPs. The TCP connection is maintained with a keep-alive flag, and seems to be reliable even on real-world cellular networks.

After establishing the data channel, the client calls the server over PSTN, which is in turn routed over VoIP using the SIP protocol. A termination-provider interfaces the phone network to the Internet for a small monthly fee. Using the voice channel instead of streaming provides a voice-optimized low-latency connection to the cellular phone. The 3G data channel does not have any quality of service in addition to much higher latency. The PSTN channel provides perceptually instantaneous playback of the desired stream neglecting latency accrued through the data channel.

The server also offsets the computational costs from the handset, calculating personalization and layout of the chat space. It interfaces with GraphViz (Ellson, 2003) to perform the energy-minimizing layout in a scriptable and flexible environment.

All data channel communication is relayed from xLink, a custom message routing server developed for the eLens. xLink buffers the potentially unreliable data connection coming from the phones, in addition to providing interfaces for other services. For example, the social networking service for the eLens, named Constellation, interfaces directly with xLink. This allows manipulation of the service from the handset without requiring awareness or additional code in RadioActive. However, Constellation can talk to RadioActive through xLink to learn about chat space activity within a given social network.

The server is written in Python on top of Twisted asynchronous server framework (Kinder, 2005). Twisted simplifies server development by providing a single-threaded non-blocking

event-based interface to the network, databases, and application state. Python allows for easy prototyping as well as the ability to bind to most other languages. Before concretely choosing a VoIP interface, it was felt that this flexibility would be important. Eventually, a SIP stack written in Python called Shtoom was chosen as the voice framework (Baxter, 2004). Shtoom provided the shell to create efficient and scalable voice channels, and natively uses Twisted. As a small open source project, Shtoom lacked sufficient documentation and quality assurance to be easily pluggable. However, because it provided the necessary function and was open source, I spent a lot of time rummaging through the code and fixing random bugs as they crept up. All bug fixes were submitted back to the community, including work to wrap the excellent iLBC codec to reduce bandwidth consumption and manipulate the sound bytes live. I ended up with a complete VoIP interface that easily allows developers to add PSTN and SIP-based audio to any back-end application.

6. EVALUATION

6.1 INITIAL EVALUATION

The first built version of RadioActive used the entire desktop as an initial testing ground for the design. It was spatially grounded, requiring users to start new threads at arbitrary points in the space. Threads were surrounded by boxes to help visually group them (Figure 6). The initial test comprised of eight males and two females. They were given a short explanation of how the chat space worked and were then asked to join in the current discussion. They could use it as long as they liked, averaging approximately 3 minutes. This was the amount of time it took to browse the small chat space and leave a reply. Afterwards they participated in an open-ended interview.

All of the subjects found the concept of asynchronous audio intriguing, imagining scenarios ranging from collaborative work to blog activities. One said it was “the next obvious step from message boards”. Nine of the subjects found the visual representation and its associated navigation intuitive, with comments ranging from “I can’t see how else you’d do it” to “once you get used to it, it makes sense”. Three subjects were concerned with how the design might translate to mobile usage where users might need to navigate without looking at the screen, such as in an urban context. However, the most contentious subject was browsing. The results were split between users who wanted one unique subject per message and those who preferred a single subject for the entire thread. Eight of the subjects thought text-based summaries or tags would help in understanding the chat space. Requiring the user record a short subject regardless of message length proved to be confusing for the users, as many left short messages and did not feel the need separate recording. They often repeated the same message over again, assuming it was an error. For the longer messages, the recorded subjects were often of poor descriptive quality, or summarized a viewpoint as “I disagree”.

It is possible that the users who preferred a separate subject for the thread did so because of the spatial organization of the chat space. As previously stated, this study illustrated how it is confusing to place a thread seemingly arbitrarily in 2D space. In future iterations

where threads were not spatially related, one would have a different starting place before launching a thread. Such a view is different from seeing a collection of messages linked with no indication of direction. However, it was felt that both views had merit and a balance could be struck for future iterations. Future designs still retained a unique subject for each message, but only when the message itself was of short duration (under ten seconds). Threads themselves were given a summary beyond the initial message in the form of text tags in a separate space.

However, the problem of acquiring usable summaries remained. This perhaps could be solved with better interface prompts, but more likely, it comes from the user's lack of experience and the lack of netiquette for the medium. I hypothesize the more RadioActive is used, usage patterns will self-regulate the quality of a subject because the value and cost of good and bad subjects, respectively, is quite high. Unfortunately, such a claim requires a massive long-term user study, beyond the scope of this thesis.

Another contentious issue was the notion of hearing range that provided the function of parallel browsing. As the user moved their mouse around, any subject that fell within the rectangular box that surrounded the cursor was previewed. The volume associated with the playback of the subject was inversely proportionate to the distance from the center of the cursor. Each user reported that they found the playback of simultaneous subjects jarring and confusing, even though they were informed of the functionality a priori. Parallel browsing was removed from future iterations, relying on single message presentation and skimming. Skimming was reused the previous concept of hovering over messages, except using touch instead of a mouse pointer.

6.2 ELENS

With the results of the previous study, further iterations lead to the design of the next tested version. This version was mainly described in chapter three. I will outline in brief the project, how it extends RadioActive, and the lessons learned. It is the principal investigation of RadioActive.

The user study was conducted in Spain in conjunction with their government for the eLens project. The eLens project sought to better link citizens and government using mobiles and physicality. It used RadioActive as a conduit for digital town meetings. The meetings would take place virtually, but physically situated in the immediate context. The physical context is acquired by scanning a unique physical tag with the phone's camera. The tags were coded to be robust (Costanza, 2003). Using the 3G data network, the client would identify the tag to the server, which in turn selects the corresponding chat space.

The tags themselves are printable from a web page and can be affixed to a building using blue tac. This setup provides a nice feedback loop between the digital and the physical that cannot be provided through purely digital localization services. Physical tags also inherit social context from their physical placement, which in turn is inherited by the digital conversation. This is a true affordance of mobility and cannot be replicated on the desktop. For example, a tag placed outside of a bar, inside behind the counter, on a table, and in the bathroom all inherit differing social context even though GPS would identify the entire location as a single unit--bar.

The project also segmented the chat spaces according to specified social networks, which are either programmed in explicitly or gleaned from physical proximity. This functionality permits a social-centric model towards dynamic alternative channels for purely civilian conversation, both public and private. It also is a mechanism to navigate through chat spaces according to audience. Using Bluetooth scanning and other techniques for acquiring social networks, RadioActive is afforded better measurements of social relevance and consequently predictive power through mobility--another augmentation that cannot be replicated on desktops.

In order to test the functionality of the eLens, including RadioActive, we collaborated with a local technical high school in Manresa, Spain to gain a set of 9 males and 6 females aged 15-19. Each student was loaned a Motorola A1000 with a SIM card provided by Telefonica with unlimited data and voice for a week. They were encouraged to use it on their own, in addition to the two-hour workshops with our team that took place during the test period. The workshops were constructed to have the test subjects play with the system under supervision to ensure they understood how the interface worked, observe their navigational strategies, answer any questions, and keep the system in working order.

6.2.1 SYSTEM PROBLEMS

Like most initial real world tests, the students quickly found bugs in the system. These would be addressed after the workshop, resulting in a new version of the software to upload to the handsets the following day. Unfortunately, the security-limitations of the handset prevented us from auto-updating the clients ourselves.

One of the main headaches beyond our control was the real world performance of a 3G network still in its infancy. The site of the user test was Manresa, a small provincial town north of Barcelona. Due to its size and location, there were few cell tower units placed within the city. When all the users would start trying to use the system concurrently in the same room, the tower would get overloaded and often momentarily drop handsets back to 2G from 3G. This resulted in dropping, rather than pausing, the voice channel in prefer-

ence for data. The voice channel could only be restored by re-establishing the phone call. This step required user confirmation due to the security model of the handset. Noticing this trend, we adapted the software to only keep a voice channel for the duration of entering a chat space in hopes of lessening the load. However, most of the time spent in the application was in the chat space so this actually did little. Because the data channel remained active, we were at least able to bring up a screen when we detected the dropped call, asking if they would like to re-establish it.

The fatal bugs and inconsistent network performance degraded the perception of the application and caused it to be only used during class. At night, the students preferred to maximize the existing functionality of the smartphone and the unlimited calling plan, using as much as 150 euros a day in bandwidth, SMS, and voice. As these existing services are too expensive for the youth to use as much as they want, RadioActive and the rest of the eLens were low priority in their minds. They used otherwise expensive synchronous video chat with each other more than any other feature of the phone.

6.2.2 TASK STRUCTURE

The actual class assignment was to act as a proxy for the local tourist office, placing tags on historic buildings and annotating them using RadioActive. It was hoped that their interaction would encourage alternative uses of tool, and discussion amongst each other. Instead they followed the directions too literally. The students simply placed the tag, scanned it in, named it, created a thread, and left a root message that was read verbatim from the official tourist pamphlet. They played back the message to make sure it was recorded, and moved to the next location on their list.

Even though the students only used the eLens unimaginatively in class, there was enough crash-free usage to ethnographically observe their usage patterns and gain insight into the interface and conceptual successes and failures. For example, as was just pointed out, students played back messages immediately after creation to confirm their existence. In the last iteration this was addressed by centering on the new post in the message-view and automatically starting playback.

6.2.3 WHAT WORKED

As a whole, the interface was successful. As was observed and reported in informal interviews, users on average took about 3-4 tries of scanning a tag, naming it, creating a thread, recording a message, and listening to it to feel that they were comfortable with the interface. Afterwards, they effortlessly traversed chat spaces, carefully moving around

the 2D plane and scanning the messages. The students that were originally adverse to new technology came around in the end. In asking for a single student who did not like the design, there was not a single volunteer. They also liked the integration of physicality and chat, as noted, “I can really see this as a possible future of communication. I think its powerful that I can leave a message on a location. We could see being late and my friend knowing where I went.” Another liked persistence, when noting how it functions “like an internet forum, I like the idea of the persistent conversation, that messages would stay for 1 year or longer. I think it’s a great way to talk to lots of people and to meet new people who sharing my interests.” In free-form interviews, the students said the interface was intuitive, and almost all liked using voice (13/15). They also expressed interest in adding videos or pictures to the chat space. The experience of voice versus SMS was described as “more personal” and “easier”, although there were concerns over the speed and cost of the phone to function equivalently. Most of the students said they imagined it being used as a tourist guide--a product of the experiment itself--but several mentioned it would be useful to leave messages for friends in social situations, such as at cafes or bars. Several also thought of it functioning like Citysearch, where users could rate and talk about a business outside of it. The only complaint about the interface was the lack of author names directly on the post. The student wanted the ability to find messages from specific people. This request was fulfilled in the follow-up design.

6.2.4 LESSONS LEARNED

RadioActive’s visual design was found to have had two main problems: the thread-screen and subjects versus messages. The problems with the thread screen were discussed above, but ultimately came down to three classes: separation between object and action, occlusion of topic in favor of message count visualization, and a lack of understanding relevance in the message count visualization. The object/action problem was corrected in the next iteration by assigning a default action to the double click of each object. It is also the case that the thread visualization may not have correctly served the user’s purpose because it gave visual priority to a user’s history within a thread. However, this preference can be explained by the goal of finding specific threads by summary rather than casually browsing, which was the expected goal. By moving away from this view completely, we achieve both goals in the new the thread-level view by specifying summary, message count, and popularity together.

The users also did not understand the purpose and interaction of messages that were split into subject and body. When prompted for a summary, they often repeated the same message assuming the original did not go through. It is likely this behavior was due to a language problem, as the software was written in English while the students mostly spoke Catalan. It is also possible that another source of confusion was the tendency to leave short messages. Since most messages were five to seven seconds, the extra prompt for the

rare longer message was foreign and unexpected. Users then assumed there was a problem because when they touched the circle it only played the subject.

The original design required users to tap a message to hear its subject, and tap again during subject playback to hear the body. This mechanism was motivated by the preference of browsing subjects over content since the bodies were the most expensive bit. However, with no visual feedback or other indication the mechanism became opaque. The next iteration corrected this by incorporating the presence of a subject & body in the circle representation (Figure 2), text-based status bars that tell which part is playing, and changing the touch-based strategies to prefer bodies. It is now the case that only pressing an individual message will play the body, while dragging a finger across messages or holding a single message for 300ms will preview the corresponding subject. Thus we use subjects as a power-user feature. Further investigations are necessary to understand the best strategy.

An unexpected observation was the use of the stylus. The interface was carefully designed to permit finger-based interaction on the touch screen, yet each time the students chose the stylus even after being told it was not necessary. The behavior may be rooted in the need for a stylus throughout the rest of the phone's software. Other explanations include the dedication of the task in terms of time allocation, low cognitive load requirements, and embracing of the touch screen for its novelty.

Another unforeseen consequence, but retrospectively obvious, was the problem of street noise when using speakerphones in urban situations. While RadioActive was originally envisioned using Bluetooth headsets, speakerphones provide a more accessible and cheaper design. Future work should consider DSP treatments to help remove excessive urban noise, should headsets not be more adapted.

Aside from the expected bugs that are only exposed in mass testing, RadioActive's main technical design flaw was its use of dual-data and voice channels. The amount of dropped calls was unexpected and due to network conditions rather than faulty programming. Future revisions should move to data-only model, using smart caching techniques to balance latency, memory consumption, and choice of which messages to buffers. By only using the data channel, the connections would be more robust, avoiding interruption from intermediate network drops.

The other technical problem was execution speed. Long loading times (30 seconds) are a consequence of the slow bus and memory speeds of the phone, in addition to its security model which forces class verification at run-time. These issues were beyond our control and are a consequence of the current state of the industry and Motorola's engineering

decisions. However, run-time performance once in the chat space was acceptable.

7. FUTURE WORK

Long-term studies are needed to evaluate the true potential of asynchronous persistent audio as a medium. However, we can think of design improvements without first knowing. The foreseen future directions for RadioActive fall into three categories: 1) immediate changes in visualization, 2) some helpful uses of ASR, and 3) adapting to mobility.

7.1 IMMEDIATE INTERFACE CHANGES

The basic chat space with the circles seems to mostly work. What really needs further evaluation is the separation of subject/message, starting with the changes that reflect the eLens evaluation. If they are understood and liked, we should vary the timing required to count as a subject. Another tactic is to have a manual preference for subjects or bodies.

While using space to separate topics automatically was shown to be a bad idea, allowing users to shuffle around messages might be useful. A clustering strategy might emerge that solves some unforeseen problem, only because the representation was malleable. Groups of messages could be selected by lassoing desired plurality and dragging them to their new location.

During playback, we can present contextually relevant hyperlinks to alternative views. A useful starting set would be 1) a list of other message by this author, 2) other messages of similar topic, and 3) an interface to understand and access the messages of the people who listened to the message. These links help the user non-linearly explore all conversation. Back/forward buttons, like on a web browser, are logically necessary to promotes exploration.

7.2 ASR

The appropriateness of ASR is dependent on the message content and user's goals. For

example, we have assumed a casual chat environment where users leave unimportant short messages. Summarizing them helps navigation, but listening to the author is part of the experience. This usage type encourages the device to be worn passively on autoplay, which could only use ASR to control navigation. However, if RadioActive were to support an engineering knowledgebase, one would only want to use just long enough to answer a specific question. Full text transcription would be useful to find the answer.

ASR would be useful in message creation, regardless of content. Implementing a feature similar to ScanMail would allow the user to semantically edit their recording as if it was in text (Whittaker, 2004). This would allow people to correct mistakes in their recording without re-recording the entire message. Because the message is finally presented in audio, the erroneous nature of ASR is tolerated.

7.3 ADAPTING TO MOBILITY

In my opinion, watching the students scan the building and listen to a message was very natural. Despite temporary problems, such as boot-up or voice channel acquisition, using space as a navigational aid makes a lot of sense on a mobile. It gives special meaning for the conversation to take place on a constrained device instead of on the desktop at home.

Space can be used as a hyperlink, augmenting everyday objects and our environment with social function through the power of mobility. We have already accomplished this through tag recognition (Costanza, 2003), but other localization techniques exist with different design traded-offs. Some common examples are computer vision (Fritz, 2004), GPS, and IR (Merrill, 2005). More unique ways of hyperlinking might involve using the surrounding people and social context. Mobiles uniquely afford integrating the physical with the digital through sensors and networked intelligence.

8. CONCLUSION

Asynchronous persistent audio, an unexplored medium, might be the proper way to flexibly and loosely connect groups on mobile devices. Mobiles uniquely afford fuzzier and more distributed social connections by better integrating with the physical environment. However, the technology to create even a context-less persistent audio chat space does not previously exist. This thesis focused on the key problem of navigating asynchronous persistent audio under the practical constraints of a mobile device. It contributes the unique ideas of a visualized audio chat space, alternative thread visualizations, persistent group chat on mobiles, automated playback of multi-threaded chat, using listening time to passively derive moderation values, and augmenting a physical location with a persistent chat. By creating, iterating, and validating a basic design, potential for the medium has been demonstrated but not proven. Long-term usage is needed to evaluate the full potential of the medium.

9. APPENDIX

9.1 A CHRONOLOGICAL LIST OF ACCOMPLISHED MILESTONES AND FEATURES

9.1.1 *VERSION 1*

- Desktop-based, written in J2SE & PostgreSQL
- Using idea of a spatially-grounded chat space, taken from chat circles
- Spatial distance between threads indicates semantic relationships.
- Messages did not have subject/body split
- Didn't have zoom, but translatable map for the infinite space
- Visual encoding for a circle only differentiated between root and reply

9.1.2 *VERSION 2 (CONCEPTUAL SKETCH)*

- Added zoom to map representation, scaled down sizes to fit more on the screen
- Kept a zooming map
- Change of color choices
- Added visual encodings including circle around for moderation value
- Started work on new thread layout engine but keeping same idea

9.1.3 *VERSION 3 (SKETCH & PARTIAL CODE)*

- Started work on actual A1000
- Moved away from spatial-grounding towards a more standard thread setup like email/newsgroups

- Started to use J2ME Polish as a GUI, which was eventually aborted in favor of constructing Barcelona on top of TinyLine
- Created concept of the inbox / inverse RSS feed
- Moved to a standard tree to simplify layout and minimize uncertainty in terms of big visual extensions according to previous algorithm
- Tried compact alternative inbox view
- Information sliders at bottom to deal with small real estate

9.1.4 VERSION 4

- Moved away from tree into energy-minimizing layout...breaking constraints of original layout while preserving spirit and giving an even more organic look
- Simplified circles, changed colors for family effects, added outer ring for 2-level gestalt effects
- Changed alternative view to include thread level information, and minimized normal thread view
- Removed scroll bars and opted for more Google-maps style interaction with the space
- Thread screen centers on first post
- Integration with constellation becomes permanent aspect of RadioActive, providing social grouping features and ability to extend into more dynamic chat spaces easily in the future

9.1.5 VERSION 5

- Finishing touches using input from the previous version's evaluation
- Added actions onto all objects
- Created visual split between message & subject
- Centers on new posts with auto playback
- Added author name
- Removed alternative thread screen in favor a minimal approach with only popularity
- Removed callin-based queuing to initiate a VoIP session. Caller ID seems more reliable, preventing the wait times that were confusing as well as collisions & mis-matched caller ID records (as well as more potential for web-version which doesn't have a Bluetooth MAC to use)

- Added popularity
- Added zoom
- Added playback heuristic
- Added concentric circles

10. BIBLIOGRAPHY

- Arons, B. A Review of the Cocktail Party Effect. *Journal of the American Voice I/O Society*. 12 July, (1992)
- Arons, B. Hyperspeech: Navigating in speech-only hypermedia. *Proc. Hypertext*. ACM Press (1991), 133-146.
- Arons, B. SpeechSkimmer: A system for interactively skimming recorded speech. *Trans on CHI* (March 1997), 4, 1, 3-38.
- Baudisch, P., Xie, X., Wang, C., and Ma, W.I. Collapse-to-zoom: viewing web pages on small screen devices by interactively removing irrelevant content. *Proc. UIST 2004*, ACM Press (2004), 91-94.
- Baxter, A. Shtoom. <http://divmod.org/trac/wiki/ShtoomProject>. (2004).
- Bederson B., and Holland J. Pad++: a zooming graphical interface for exploring alternate interface physics. *Proc. UIST 1994*, ACM Press (1994), 17-26.
- Blattner M., Sumikawa D., and Greenberg R. Earcons and icons: Their structure and common design principles. *Human Computer Interaction* 4, 1 (1989), pp. 11-14.
- Chalfonte B.L, Fish, R.S, and Kraut, R.E. Expressive richness: a comparison of speech and text as media for revision. *Proc CHI 1991*. ACM Press (1991), 21-26.
- Connell, J. Mendelsohn, G., and Robins, R. "Effects of Communication Medium on Interpersonal Perceptions: Don't Hand Up on the Telephone Yet!", *Proceedings of GROUP '01*, Boulder, CO, (2001), pp. 117-124.
- Costanza, E., Shelley, S. B., and Robinson, J., d-touch: a Consumer-Grade Tangible Interface Module and Musical Applications. *Proc. Designing for Society HCI2003*, Bath, UK, 8-12 (September 2003)
- Donath, J., Karahalios, K., and Viegas, F. Visualizing conversation. *Proc. HIICS-32, System Sciences*, (1999)
- Dourish, P. and Chalmers, M. Running out of space: models of information navigation. Short paper presented at *HCI'94*, Glasgow. (1994)
- Ellson, J., Gansner, E.R., Koutsofios, L., North, S.C, and Woodhull, G. Graphviz and dynagraph - static and dynamic graph drawing tools. *Graph Drawing Software*. (2003)

- Fernström, M., and Brazil, E. Sonic Browsing: An Auditory Tool for Multimedia Asset Management. Proc ICAD '01, Espoo, Finland, July 2001.
- Fritz, G., Seifert, C., Luley, P., Paletta, L., and Almer, A. Mobile Vision for Ambient Learning in Urban Enviroments. Proc. International Conference on Mobile Learning, MLEARN (2004)
- Goodman, B., Geier, M., Haverty, L., Linton, E., and McCready, R. A Framework for Asynchronous Collaborative Learning and Problem Solving. Proc. AI in Education, IOS Press (2001)
- Hewitt, J. Beyond threaded discourse. Proc WebNet '97. AACE (1997).
- Hindus, D., Schmandt, C., and Horner, C. Capturing, structuring and representing ubiquitous audio. ACM Trans. on Information Systems, 11, 44 (Oct. 1993), 376-400.
- Kerr, B.J. Thread arcs: An email thread visualization. Proc Infovis '03 (2003).
- Kilander, E., and Loennqvist, P. A Whisper in the Woods-An Ambient Soundscape for Peripheral Awareness of Remote Processes. Proc ICAD '02.
- Kinder, K. Event-driven programming with Twisted and Python. Linux Journal (2005).
- Kim J.S. TattleTrail: An Archiving Voice Chat System for Mobile Users Over Internet Protocol. MIT M.Eng Thesis, May 2002.
- Kirsh, D. Metacognition, Distributed Cognition and Visual Design. Cognition, Education and Communication Technology. Lawrence Erlbaum, 2004
- Kirsh, D. The intelligent use of space. Journal of Artificial Intelligence, (1995) 73(1-2), pp. 3168.
- Lakshmiathy, V., and Schmandt, C. SimPhony, A Voice Communication Tool for Distributed Workgroups. BT Technology Journal, 22, 4 (Oct. 2004), 52 – 61
- Lam D., Rohall, S.L., Schmandt, C., and Stern, M.K. Exploiting E-mail Structure to Improve Summarization. Proc. CSCW 2002, ACM Press (2002)
- Merril, D., and Maes, P. Invisible Media: Attention-sensitive informational augmentation for physical objects. UBICOMP '05 (2005)
- McGookin, D., and Brewster, S. An Investigation into the Identification of Concurrently Presented Earcons. Proc. ICAD, Boston, MA, July 6 - 2003, 42-46.
- Mynatt, E., Back, M., Want, R., and Federick R., Audio Aura: Light-Weight Audio Augmented Reality. Proc UbiComp '97. ACM Press (1997).
- Nishimoto T., Yuki, H., Kawahara, T., Araki, T., and Niimi, Y. Design and evaluation of the asynchronous voice meeting system AVM. Systems and computers in Japan, Wielely Periodicals (2002), 33, 11, 61-69.
- Ohlund, B., Yu, C.H., and Jannasch-Pennell, A.. Impact of asynchronous and synchronous internet-based communication on collaboration and performance among k-12 teachers. Journal of Educational Computing Research (2002), 23, 405-420.
- Resnick P. HyperVoice: A Phone-Based CSCW Platform. Proc. CSCW 1992, ACM Press (1992) 218-225.

- Rodenstein, R., and Donath, J. Talking in Circles: Designing a Spatially-Grounded Audio Conferencing Environment. *Proc CHI 2000*. ACM Press (2000) 81-88.
- Rohall S., Gruen D., Moody P., and Kellerman S. Email Visualizations to Aid Communications. *IEEE Symposium on Information Visualization*, 2001.
- Ross, K.W. Asynchronous voice: a personal account. *Multimedia, IEEE*, 10, 2, April-June 2003, 70-74.
- Sawhney, N., and Murphy A. ESPACE 2: An Experimental HyperAudio Environment. *Proc CHI 1996*. ACM Pres (1996), p. 105-106.
- Schmandt, C. Audio Hallway: A Virtual Acoustic Environment for Browsing. *Proc UIST 1998*. ACM Press (1998) 163-170.
- Schmandt, C., Kim, J., Lee, K., Vallejo, G., and Ackerman, M. Mediated Voice Communication via Mobile IP. *Proc. UIST 2001*. ACM Press (2001), 141-150.
- Smith, M., and Fiore, A. Visualization Components for Persistent Conversations. Microsoft Technical Report MSR-TR-2000-98.
- Tucker, S., and Whittaker, S. Time is of the Essence: An Evaluation of Temporal Compression Algorithms. In *Proceedings of CHI06 Conference on Human Factors in Computing Systems*, New York: ACM Press (2006).
- Tufte, E. *Envisioning Information*. Graphics Press (May 1990)
- Venolia, G. D., Dabbish, L., Cadiz J., and Gupta, A. Supporting email workflow. Technical Report MSR-TR-2001-88, Microsoft Research, Collaboration & Multimedia Group, One Microsoft Way, Redmond, WA 98052 USA, December 2001.
- Venolia, G., and Neustaedter, C. Understanding sequence and reply relationships within email conversations: A mixed-model visualization. *MSR-TR-2002-102*. (2002)
- Viegas, E., and Donath, J. Chat Circles. *Proc CHI 1999*. ACM Press (1999), 9-16.
- Whittaker, S., and Amento, B. Semantic Speech Editing. *Proceedings of Conference on Computer Human Interaction*, 527-534, New York, ACM Press. (2004).
- Whittaker, S., Hirschberg, J., Amento, B., Stark, L., Bacchiani, M., Isenhour, P., Stead, L., Zamchick G., and Rosenberg, A. SCANMail: a voicemail interface that makes speech browsable, readable and searchable. *Proc. CHI 2002*, ACM Press (2002), 275-282.