Designing Audio Aura

Elizabeth D. Mynatt, Maribeth Back, Roy Want
Xerox Palo Alto Research Center
[mynatt,back,want]@parc.xerox.com

Michael Baer
Stanford University
calbaer@stanford.edu

Jason B. Ellis
Georgia Institute of Technology
jellis@cc.gatech.edu

ABSTRACT
In this paper, we describe the process behind the design of Audio Aura. The goal of Audio Aura is to provide serendipitous information, via background auditory cues, that is tied to people's physical actions in the workplace. We used scenarios to explore issues in serendipitous information such as privacy and work practice. Our sound design was guided by a number of strategies for creating peripheral sounds grouped in cohesive ecologies. Faced with an physical and software infrastructure under development in a laboratory distant from our sound studio, we prototyped different sonic landscapes in VRML worlds. In our infrastructure design, we made a number of trade-offs in our use of legacy systems and our client-server design.

Keywords: Audio, Augmented Reality, Auditory Icons, Active Badge, VRML, Earcons, Awareness, Periphery

INTRODUCTION
In this work we explore using audio to connect a person's activities in the physical world with information culled from the virtual world[1]. Given the richness and variety of activities in typical offices, interaction with computers is generally limited and explicit. It is primarily limited to our typing and mousing into a box seated on our desk. Our dialogue is explicit; we enter in commands and the computer responds. The purpose of Audio Aura is to create implicit dialogues with our computers that occur away from our desk. There are three targeted constraints in our design. First, we use audio to create peripheral cues. Second, we provide serendipitous information, useful but not required. Third, we tie the delivery of information to physical actions in the workplace such as stopping by someone's office.

In Audio Aura, we use audio to provide information that lies on the edge of background awareness. We naturally use our ears to monitor our environment; when we hear someone approaching, when we hear someone say our name, when we hear that our computer's disk drive is spinning. While in the midst of some conscious action, our ears are gathering information that we may or may not need to comprehend. Audio, primarily nonspeech audio, is a natural medium to

1. Virtual world refers to cyberspace, information in networked computer systems.

create a peripheral display. Our goal is to leverage these natural abilities and create an interface that enriches the physical world without being distracting.

The information we provide via Audio Aura is designed to be serendipitous. You appreciate it when you hear it, but you do not rely on it in the same way that you rely on receiving an email message or a message reminder. The reason for this distinction should be clear. Information that you rely on must invade your periphery to ensure that it has been perceived. This limitation does not imply that serendipitous information is not of value. Conversely, many of our actions are guided by the wealth of background information in our environment. Whether we are reminded of something to do, warned of difficulty along a potential path or simply provided the spark of a new idea, our opportunistic use of serendipitous information makes our lives more efficient and rich. Our goal is to provide useful, serendipitous information in our workplace.

Computers are not particularly well designed to match the variety of activities that are part of being a physical human being. We walk around. We get coffee, the mail, our lunch. We go to conference rooms. We drop by people's offices. Although some computers are now small enough to go with us, they don't take advantage of our physical actions. In Audio Aura, our goal is to leverage everyday physical activities. One opportune time to provide serendipitous information on the periphery is when a person is walking down the hallway. If the person is concentrating on their current task, they will likely not notice or attend to the display. If, however, they are less focused, they should naturally notice and perhaps decide to attend to information on the periphery. Additionally, physical actions can guide the information content. A pause at someone's empty office is an opportune time to hear whether they have been in the office earlier that day.

In summary, the goal of Audio Aura is to provide serendipitous information, via background auditory cues, that is tied to people's physical actions in the workplace. Our current system combines three known technologies: active badges, distributed systems and digital audio delivered via portable, wireless headphones. An active badge [10] is a small electronic tag designed to be worn by a person. It repeatedly emits a unique infrared signal that is detected by a low-cost network of IR sensors placed around a building. The information from the IR sensors is collected and combined with other data sources such as online

calendars and email queues. Audio cues are triggered by changes in this Audio Aura database and sent to the user's wireless headphones.

System goals are, first, to be able to provide multiple sources of information content as well as multiple means for triggering the delivery of the information. Second, the system should be easily configurable by end users because information needs and interface preferences will vary between users. Third, services using Audio Aura should be easy to author and lightweight to run.

The Design Process

In this paper, we describe the process behind the design of Audio Aura. During the course of this design we attempted to address the kinds of information that could be presented via Audio Aura, the expressiveness and aesthetics of the auditory cues and the utility and flexibility of the underlying infrastructure. In the following section, we describe three sample scenarios of serendipitous information that have guided our design. These scenarios highlight issues in the responsiveness of the system, privacy and the complexity of the information sources. We then turn to the design of the individual auditory cues and strategies for presenting these sounds. During this design we have explored the use of different types of sounds: speech, musical and sound effects.

Designing for a distributed physical environment where the underlying Audio Aura infrastructure was still under development presented several challenges. We decided to use a virtual reality environment to prototype and explore different designs. Although this virtual representation has been useful for a number of reasons, we've learned some lessons regarding transitioning designs from the virtual prototype to the physical world.

The computational and hardware infrastructure for Audio Aura is based on a legacy system for ubiquitous computing [11]. Our initial plans for building directly on top of this infrastructure were overly optimistic. Although sufficient for the visually-oriented, non-interactive applications that were created with the original system, the infrastructure for detecting the location and movements of the users as well as scripting complex responses to those actions needed to be adapted to our requirements. We describe the modifications that we made to the hardware components and the creation of software infrastructure for programming Audio Aura services. These services can easily gather and store sources of data used to trigger the display of auditory cues.

Related Work

The design of Audio Aura has been inspired by several related efforts. Most work in augmented reality systems [2][4] has focused on augmenting visual information by overlaying a visual image of the environment with additional information usually presented as text. A common configuration of these systems is a hand-held device that can be pointed at objects in the environment. The video image with overlays is displayed in a small window. These handheld systems require the user to actively probe the environment as well as indirectly view a representation of the environment on the video screen. Our system offers two primary distinctions. First, users do not have to actively

probe the environment. Their everyday pattern of walking throughout an office environment triggers the delivery of aural information. Second, users do not view a representation of the physical world, but continue to interact with the physical world that includes additional real-world auditory cues. This lack of indirection changes the experience from analyzing the physical world to participating in the physical world.

Providing auditory cues based on people's motion in the physical environment has also been explored by researchers and artists, and is currently used for gallery and museum tours. The systems that most closely approach ours include one described by Bedersen [2], where a linear, usually cassette-based audio tour is replaced by a non-linear, sensor-based digital audio tour allowing the visitor to choose their own path through a museum. Several differences between our systems are apparent. First, in Bedersen's system users must carry the digital audio data with them, imposing an obvious constraint on the range and generation of audio cues that can be presented. Second, Bedersen's system is unidirectional. It does not send information from the user to the environment such as the identity, location, or history of the particular user.

Other investigations into audio awareness include Hudson [7] who demonstrated providing iconic auditory summaries of newly arrived email when a user flashed a colored card while walking by a sensor. This system still required active input from the user and only explored one use of audio in contrast to creating an additional auditory environment that does not require user input.

Explorations in providing awareness data and other forms of serendipitous information illustrate other possible scenarios in this design space. Ishii's Tangible Bits [8] focuses on surrounding people in their office with a wealth of background awareness cues using light, sound and touch. Our work follows the user outside of their office where their activities trigger different awareness cues. Gaver et al [6] explored using auditory cues in monitoring the state of a mock bottling plant. Pederson [9] has also explored using awareness cues to support awareness of other people.

SERENDIPITOUS INFORMATION

Based on informal observation of our colleagues we devised three scenarios of use for Audio Aura that guided our design. These scenarios touched on issues in system responsiveness, privacy, and the complexity and abstractness of the information presented. Each scenario grew out of a need for different types of serendipitous information. First, we are an email-oriented culture[1]. Whether we have newly-arrived email, who it is from, and what it concerns is often important. People will run by their offices between meetings to check on this important communication pipeline. Another common between-meeting activity is dropping by the bistro to get a cup of coffee or tea. One obvious tension is whether to linger with your coffee and chat with colleagues or to go check on the latest email messages. We decided to tie these activities

1. The main exception is managers who shift to using voice mail.

together. When you enter the bistro, you will hear a cue that conveys approximately how many new email messages you have and indicates messages from particular people and groups.

Second, people tend to opportunistically drop by people's offices. This practice supports communication when an email message or phone call might be inappropriate or too time consuming. When an opportunistic visitor is faced with an empty office, they may quickly survey the office trying to determine if the desired person has been in that day. In Audio Aura, the visitor now hears an auditory cue conveying whether the person has been in that day, whether they've been gone for some time, or whether you just missed them. It is important to note that these cues are *qualitative*. They do not report that "Mr. X has been out of the office for two hours and 45 minutes." The cue gives a sense akin to seeing their light on and their briefcase against the desk or hearing a passing colleague report that the person was just seen walking toward a conference room.

Third, many people are not co-located with their collaborators. These people often do not create and share a palpable sense of their group's activity analogous to one shared by a co-located group. In this scenario, various bits of information about individuals in a group become the basis for an abstract representation of a "group pulse." Whether people are in the office that day, if they are working with shared artifacts, or if a subset of them are collaborating in a face-to-face meeting triggers changes in this auditory cue. As a continuous sound, the group pulse becomes the backdrop for other Audio Aura cues.

SOUND DESIGN

In this section we discuss the design issues related to constructing sounds for Audio Aura. We created several sets, or ecologies, of auditory cues for each of the three scenarios. Each sound was crafted with attention to its frequency content, structure and interaction with other sounds. To explore a range of use and preference, we created four sound environments composed of one or more sound ecologies. The sound selections for email quantity and the group pulse are summarized in Tables 1 and 2.

Design intent: sonic ecologies

Because we intend this system for background interaction, the design of the auditory cues must avoid the "alarm" paradigm so frequently found in computational environments. Alarm sounds tend to have sharp attacks, high volume levels, and substantial frequency content in the same general range as the human voice (200 – 2,000 Hz). Most sound used in computer interfaces has (sometimes inadvertently) fit into this model. We are deliberately aiming for the auditory periphery, and our sounds and sound environments are designed to avoid triggering alarm responses in listeners.

One aspect of our design approach is the construction of sonic ecologies, where the changing behavior of the Audio Aura system is interpreted through the semantic roles sounds play. For example, particular sets of functionalities can be mapped to various beach sounds. In our current sound effects design, the amount of email is mapped to

seagull cries, email from particular people or groups is mapped to various beach birds and seals, group activity level is mapped to surf, wave volume and activity, and audio footprints are mapped to the number of buoy bells.

Another idea we are exploring in these sonic ecologies is imbedding cues into a running, low-level soundtrack, so that the user is not startled by the sudden impingement of a sound. The running track itself carries information about global levels of activity within the building, within a work group, or on the network. This "group pulse" sound forms a bed within which other auditory information can lie.

Structure of individual sounds

One useful aspect of the ecological approach to sound design is considering frequency bandwidth and human perception as limited resources. Given this design perspective we must build the sounds with attention to the perceptual niche in which each sound resides.

Within each design model, we have tried several different types of sounds, varying the harmonic content, the pitch, the attack and decay, and the rhythms caused by simultaneously looping sounds of different lengths. For example, by looping three long, low-pitched sounds without much high harmonic content and with long, gentle attacks and decays, we create a sonic background in which we leave room for other sounds to be effectively heard. In the music environment this sound is a low, clear vibe sound; in the sound effects environment, it is distant surf. These sounds share the sonic attributes described above.

Types of aural environments

The Audio Aura system offers a range of sound designs: voice only, music only, sound effects only, and a rich sound environment using all three types of sound. These different types of auditory cues, though mapped to the same type of events, afford different levels of specificity and required awareness. Vocal labels, for example, provide familiar auditory feedback; at the same time they usually demand more attention than a non-speech sound. Because speech tends to carry foreground information, it may not be appropriate unless the user lingers in a location for more than a few seconds. For a user who is simply walking through an area, the sounds remain at a peripheral level, both in volume and in semantic content.

MODELING SONIC BEHAVIORS IN VRML 2.0

The Audio Aura augmented reality system is tied to the physical infrastructure of the Computer Science Lab (CSL) where the IR sensors are installed. However the sound studio where the auditory cues are designed is several halls away. This arrangement presented a logistical problem for hearing our developing sound designs. Additionally, we wanted a design environment that allowed us to hear our sonic ecologies in development. Our requirements for such an environment were:

- Ability to play multiple high-quality sounds at once, with differing behaviors

- Ability to mimic the behavior of the Audio Aura system

- Ease of translation to real system.

TABLE 1. Example of sound design variations between types for email quantity

	Sound Effects	Music	Voice	Rich
Nothing new	a single gull cry	high, short bell melody, rising pitch at end	"You have no email."	Same as SFX: a single gull cry
A little (1 - 5 new)	a gull calling a few times	high, somewhat longer melody, falling at end	"You have n new messages."	a gull calling a few times
Some (5 - 15 new)	a few gulls calling	lower, longer melody	"You have n new messages."	a few gulls calling
A lot (more than 15 new)	gulls squabbling, making a racket	longest melody, falling at end	"You have n new messages."	gulls squabbling, making a racket

TABLE 2. Example of sound design variation for group pulse

	Sound Effects	Music	Voice	Rich
Low activity	distant surf	vibe	none	combination of surf and vibe
Medium activity	closer waves	same vibe, with added sample at lower pitch	none (we tried breathing, but it was obnoxious)	combination of closer waves and vibe
High activity	closer more active waves	as above, three vibes at three pitches and rhythms	none	combination of waves and vibe, more active

We chose to use VRML 2.0 [1], a data protocol that allows realtime interaction with 3D graphics and audio in Web browsers. Mapping Audio Aura's matrix of system behaviors to a multi-layered sound design has been greatly aided by these prototyping efforts. By moving through a 3D graphical representation of the CSL offices and triggering audio cues either through proximity or touch, the sound designer gets a sense of how well the sounds map to the functionality of the Audio Aura system, and how well the sounds work together.

Four sound designs in VRML prototype

We used the same 3D model and sensor set to realize four different sound designs in our VRML prototypes:

- **Voice world:** Vocal labels on the doorway of each office and area give the room's name or number, e.g., "CSL Library" or "2101." These labels are designed as defaults and are meant to be changed by the room's current occupant, e.g., "Joe Smith." This environment was useful for testing how the proximity sensors and sound fields overlapped (see Figure 1) as well as exploring using Audio Aura as a navigational aid.

- **Sound effects world:** This design makes use of the "auditory icon" [6] model of auditory display where meaning is carried through sound sources. This soundscape is a beach, where group activity is mapped to wave activity, email amount is mapped to amount of seagull calls, particular email senders are mapped to various beach animals such as different birds and seals, and office-occupancy history (audio footprints) is mapped to buoy bells.

- **Music world:** This design makes extended use of the "earcon" [3] model of auditory display, where meaning is carried through short melodic phrases or musical treatments. Here, the amount of email is indicated by the changing melodies, pitches, and rhythms of a set of related short phrases. The "family" of email-quantity sounds consisted of differing sets of fast arpeggios on vibes. A different family of short phrases, this time simple, related melodies on bells, are mapped to audio footprints. Again, though the short melodies are clearly related to each other, the qualitative information about office occupancy is carried in each phrase's individual shifts in melody, rhythm, and length. Finally, a single low vibe sound played at different pitches portrays the group activity level. One aspect of the use of earcons is that they do require some learning; both of which family of sounds is mapped to what kind of data, and within each family, what the differences mean. In general, we opted for the simplest mappings, e.g., more (notes) means more (mail).

- **Rich world:** The rich environment combines sound effects, music, and voice into a rich, multi-layered environment. This combination is the most powerful because it allows wide variation in the sound palette while maintaining a consistent feel. However, this environment also requires the most careful design work, to avoid stacking too many sounds within the same frequency range or rhythmic structure.

Replicating sensor and system behavior

The design of the sensor arrays in the VRML prototype worlds does not exactly replicate the IR sensor network in
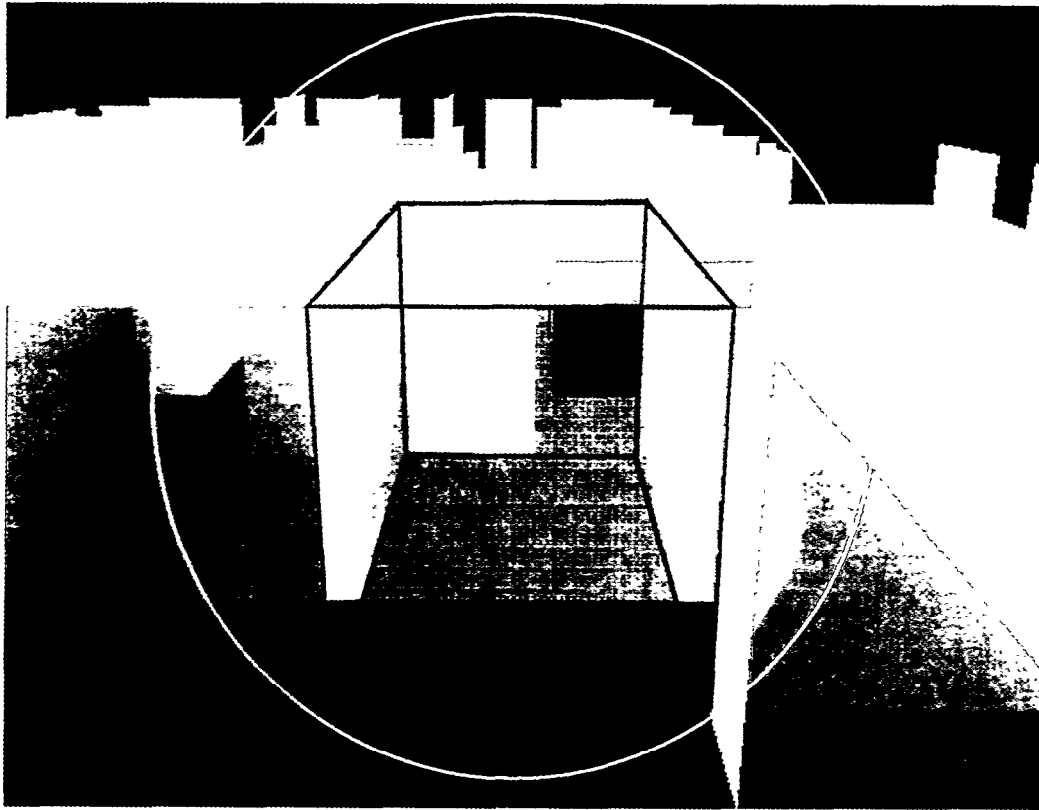
Figure 1: VRML sensor and sound geometry. The box shows the proximity sensor for the inside of this office model; the sphere shows the accompanying sound ellipse.

Each office has such a system both for its interior and for its door into the hallway.

the CSL office space. We first considered noting the physical location of each real-world IR sensor and then creating an equivalent sensor in the VRML world. However, the characteristics of the VRML sensors as well as the characteristics of VRML sound playback were not compatible with this design model. For example, the real IR sensors often require line-of sight input and the wireless headphones do not have a built-in volume mapping to proximity[1]. Because our intent in building these VRML prototypes was to understand the sonic behavior of the system, we aimed to build a set of VRML sensors and actuators that would reasonably approximate rather than replicate the behavior of the IR sensors and the Audio Aura servers. We needed to know who the user was, where the user was and at what time, within a granularity of a few feet; and we needed to be able to play sounds based on that information. We found that VRML 2.0 performed this function well.

Transition from virtual system to real system

The same sets of sounds that we use in the VRML prototypes were loaded directly into the Audio Aura services. Some performance parameters differed, like lag time before sound playback and accurate sensing of

---

1. If you're walking away from a sound's location, it doesn't automatically diminish in volume, as it typically does in VRML.

position. However we had anticipated these differences in the systems and the sounds were designed to allow a certain amount of "play."

AUDIO AURA INFRASTRUCTURE

The infrastructure for Audio Aura is comprised of legacy systems taken as is, modified legacy systems and new infrastructure built for Audio Aura. The legacy systems, as described in the following section, are the active badge system for determining people's locations in the building and the location server that collates this location information into one centralized data store. The active badge network was modified to make it more responsive fulfilling Audio Aura's need for quasi-realtime interaction. The location server, written in Modula-3 was taken as is, but new infrastructure was built to create a richer data store that supports more complex queries. This new piece, the Audio Aura server, is used by multiple Audio Aura services. For example, one of these thin clients is used for implementing each of the three scenarios. We have created a service base class in Java that facilitates the easy authoring of Audio Aura services.

Active Badges

The active badge system [10] was designed to track the locations of people in a large office building. The system operates on the premise that a person wishing to be located wears a tag called an active badge. The badge emits a
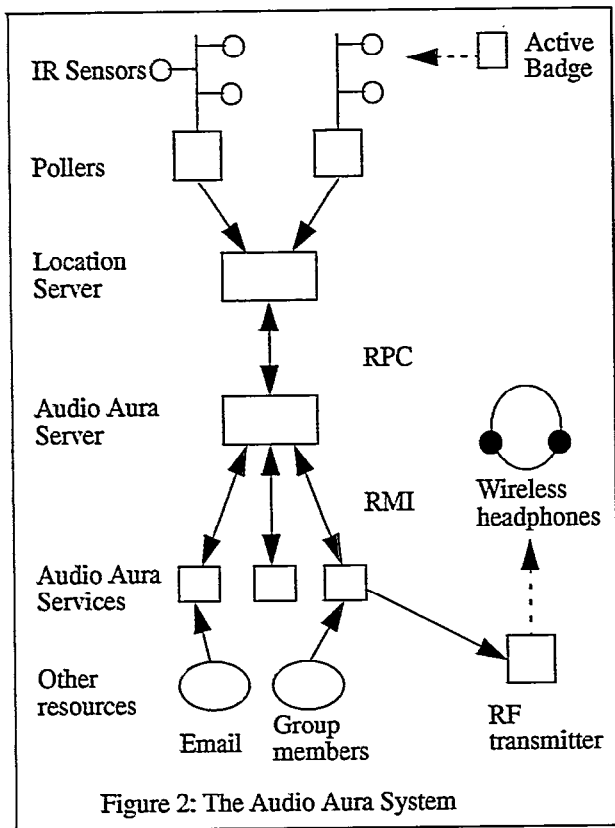
Figure 2: The Audio Aura System

unique digitally-coded infrared signal that is detected by a network of sensors, approximately once every 15 seconds. Each sensor monitors a room and detects badges up to 25 feet away. Larger rooms contain multiple sensors to ensure good coverage. Badge signals that are received by a sensor are stored in a local FIFO memory. A sensor has a unique network ID and is connected to a 9600 baud wired network that is polled by a master station (called the Poller). When a sensor is read by the Poller it returns the oldest badge sighting contained in its FIFO and then deletes it. This process continues for all subsequent reads until the sensor indicates its FIFO is empty, at which point the poller begins interrogating a new sensor. A poller collects information that associates locations with badge IDs and the time when they were read.

In our system a poller is a computer that is connected to an office LAN. A large building may contain several networks of sensors and therefore several pollers. To provide a useful network service that can be accessed by clients, the Poller information is centralized in another entity we call the location server. The location server processes and aggregates the low-level badge-ID/location-ID data and resolves the information as human understandable text. Queries can be made on the location server in order to match a person, or a location, and return the associated data. The location server also exports a network interface that allows other network clients, such as the Audio Aura system, to use the information

A design assumption implicit in the original set of active badge applications is that people generally spend much

more time stationary than in motion and when they do move, it is not at any great speed. Specifically, none of the originally envisioned applications required interaction that would quickly follow a change in location. The relatively slow badge beacon rate and the use of a polled network are both a direct result of this assumption and supported trade-offs in various engineering issues such as power consumption and signal contention.

However, for Audio Aura applications a more responsive system is required. The timely delivery of audio signals to a user at a specific location is essential to the operation of Audio Aura. In order to extend the system so that Audio Aura could make use of the active badge system, we modified some of the system components. First, we decreased the beacon period of the active badges to about 5 seconds. This increased frequency results in badge locations being revealed on a more regular basis but increases the likelihood of signal collision. At this stage of Audio Aura's development with a few users of the prototype system the increased collision probability has not been problematic. Second, we increased the speed of the polling cycle removing any wait states in the polling loop. In fact, a more critical factor than the self-imposed delays were delays caused by the polling computer sharing its cycles with other processes and tasks. We have recently dedicated a whole computer to the sole task of polling.

The active badge system has been used to provide the main source of data that triggers audio delivery in the Audio Aura system. Another legacy system that forwards keyboard activity to the location server is also used. Users permitting this keyboard information to be propagated are identified to be in their office when they are typing. As the system progresses and provides more utility we plan to combine many sources of location and activity information making use of the strengths that each system brings, thus optimizing the responsiveness and generality of Audio Aura.

Server and Services

The new pieces of infrastructure built for Audio Aura are:

- **Audio Aura Server** - This is the nerve center for Audio Aura. Written in Java it communicates to the location server via RPC. In contrast to the location server, it can store data over time and respond to general queries.

- **Audio Aura Services** - Written in Java, these services tell the server (via RMI) what data to collect and provide queries to run on that data. When queries get hits, the server returns results to the appropriate service. The services use this information as well as data from other sources to trigger the presentation of auditory cues.

The system is fully client/server with relatively thin clients. Most of the computation occurs within the Audio Aura Server. This centralization reduces network bandwidth as the server need not update multiple data repositories each time it gets new data. The server only sends data over the network when queries produce results. This technique also reduces the load on client machines. So far, the delay between the clients and the server has been negligible compared to delays in the legacy system.

Audio Aura Server The Audio Aura Server provides two fundamental extensions to the existing infrastructure at PARC — the ability to store data over time and the ability to easily run complex queries over that data. When the Audio Aura Server starts, it creates a baseline table ("csight") that is known to exist at all times. This table stores the most recent sighting for each user.

When an Audio Aura Service registers with the Audio Aura Server, it provides two things:

- **Data collection specifications:** Each of these specifications creates a table in the server. The specification includes a superkey for the table as well as a lifetime for data in that table. When the server receives new data, this specification is used to decide if the data is valid for the table and if it replaces other data.

- **Queries to run against the tables:** These queries are defined in the form of a query object. This query language provides the subset of SQL relevant to our task domain. It supports cross products and subsets as well as optimizations such as short-circuit evaluation.

After the server has updated each table with the new positioning data, it executes all the queries for services. If any of the queries have hits, it notifies the appropriate service and feeds it the results.

Services can also request an ad hoc query to be executed immediately. This type of query is not installed and is executed only once.

Audio Aura Services Audio Aura Services are relatively easy to author client processes that rely on data gathered by the server. Each service specifies the data it is interested in tracking and queries that will match interesting patterns in that data. When a service starts the data specification and queries are uploaded in the server. The service is then notified when a query gets a result.

As Java applications, these services can also maintain their own state as well as gather information from other sources. A returned query from the server may result in the service playing an auditory cue, gathering other data, invoking another program and/or sending another query to the server.

To author a service, the first step is to inherit from the service base class and override a few methods; two methods defining the data specification tables and queries, and two methods awaiting results from the server. More experienced programmers may define special initialization routines, provide a new user interface, and take advantage of some of the more complicated features of the query language.

Query Language The query language in Audio Aura is heavily influenced by the Intermezzo's database system [5]. This language is the subset of SQL most relevant to our task domain, supporting our dual goals of speed and ease of authoring. A query involves two objects:

- **AuraQuery:** The root node of the query that contains general information about the query as a whole.

- **AuraQueryClause:** The basic clause tests one of the fields in a table against a user-provided value. All clauses are connected by the boolean AND operator.

The following query returns results when "John" enters room 35-2107, the CSL Bistro. First we set the query attributes such as its ID, what table it refers to, and whether it returns the matching records or a count of the records. Next, we describes the clauses in the query by specifying field-value pairs.

```
auraQuery aq;
auraQueryClause aqc;
aq=new auraQuery();

/* ID we use to identify query results */
aq.queryId = 0;
/* current sightings table */
aq.queryTable = "csight";
/* NORMAL or CROSS_PRODUCT */
aq.queryType = auraQuery.NORMAL;
/* return RECORDS or a COUNT of them */
aq.resultForm = auraQuery.RECORDS;

/* we've seen John */
aqc=new auraQueryClause();
aqc.field = "user";
aqc.cmp = auraQueryClause.EQ;
aqc.val = "John";
aq.clauses.addElement(aqc);

 /* John is in the bistro */
aqc=new auraQueryClause();
aqc.field = "locId";
aqc.cmp = auraQueryClause.EQ;
aqc.val = "35-2107";
aq.clauses.addElement(aqc);

/* John just arrived in the bistro */
aqc=new auraQueryClause();
aqc.field = "newLocation";
aqc.cmp = auraQueryClause.EQ;
aqc.val = new Boolean(true);
aq.clauses.addElement(aqc);
```

Sound Generation

Although we are eagerly awaiting the planned additions to support sound playback in Java, we are currently not using the current Java facilities for playing sounds as it is limited to 8-bit, u-law sounds. Services currently invoke an external program to play CD-quality sounds.

CONCLUDING REMARKS

This paper summarizes the steps we have taken in designing and building the Audio Aura system. Lessons learned at each phase of the process influenced our evolving design. While we were working on scenarios, we explored other uses of Audio Aura such as delivering reminders and supporting tours and other visitors. Our discussions helped clarify the intent behind serendipitous information. We used the scenarios to constrain our sound design as well as inform our system requirements.

Our sound design was guided by a number of design strategies for creating peripheral sounds grouped in cohesive ecologies. Faced with an physical and software infrastructure under development in a laboratory distant from our sound studio, we decided to prototype different sonic landscapes in VRML worlds. As a tool for

"sketching" sound collections this tool was incredibly useful. By adjusting the behavior of proximity sensors and sound fields, we were able to minimize difficulties in transitioning our sounds to the "real world."

We faced a number of tradeoffs in our infrastructure design. We first needed to uncover and understand the original design assumptions behind the implementation of the active badge system. The remaining delays in the system influenced our design scenarios. For example, the auditory footprints service is for users who linger briefly in someone's office as opposed to users who glance in as they walk by. We would also like to move previously installed sensors to be closer to the entryway. Traditionally they are located near the center of the office. We also discussed trade-offs in various client-server designs for the Audio Aura server and services. By using a centralized data store and uploaded queries, we were able to minimize network traffic as well as complexities in writing services.

Initial User Reactions

Methods for evaluating a system designed to deliver serendipitous information in the periphery are difficult to design. How cues in the periphery are perceived and the overall value of serendipitous information is difficult to quantify. We recently demonstrated Audio Aura to nine volunteer subjects. The process involved a brief introduction followed by a set of self-paced tasks such as going to the CSL bistro, and ended with a questionnaire. Comments and questions were encouraged throughout as we were more interested in getting user feedback than performance numbers. We used the SFX (beach landscape) sound ecology. The three services (email, footprints, and group pulse) used static data so that all of the users heard the same sound cues.

Overall reactions were positive. Users found the sound choice to be good in general. They felt sounds remained in the periphery nicely although some found the meaning of sounds difficult to remember. Users thought services were well chosen and most found sound quality good. Not surprisingly, users said the time to play sounds was too long as the dedicated poller system was not available. We hope to be able to report on the long-term use of Audio Aura in the future.

Future Work

There is always a danger when computer systems are used to collect and store information on people's activities. We are currently designing mechanisms so that users can specify how data about them may be accessed. Although the existing services rely on accumulated data, the life of the data is still quite short (no more than one day). By using qualitative cues, we have also attempted to illustrate how information regarding people's activities should be presented.

We plan to create more Audio Aura services including tie-ins to voice mail and refinements on email and group activity data. We are committed to using high-quality sound in Audio Aura. We plan to integrate the high-quality Java audio engine when it becomes available late in 1997.

We are now designing an on-line user interface that allows personalization of system using the VRML prototype worlds. With this system, mapping of sounds to Audio Aura's functionalities be done either by a system designer or by the end user. The user can select sounds from a database or create their own, load them, and then test them via the VRML prototype. These VRML prototypes help users decide what pattern of sounds work best for them. The act of choosing VRML sounds will also select the sounds for the user's real Audio Aura services.

We are trying different methods of delivering wireless audio, including different types of wireless headsets and tiny in-ear monitors. Ideally, we would like to combine the IR badge and a wireless audio system into a single lightweight, non-intrusive, comfortable unit that will allow real-world sound as well as Audio Aura's sound into the ear.

REFERENCES

[1] Ames, A., Nadeau, D., Moreland, J. *The VRML 2.0 Sourcebook.* Wiley, 1996. See also the VRML Repository at http://sdsc.edu/vrml.

[2] Bederson, B.B and Druin, (In press) A. Computer Augmented Environments: New Places to Learn, Work and Play, ed. Jakob Nielsen, in *Advances in Human Computer Interaction,* Vol. 5, Ablex Press.

[3] Blattner, M. M., Sumikawa, D. A, & Greenberg, R. M. (1991). Earcons and Icons: Their Structure and Common Design Principles. *Human-Computer Interaction* 4(1), 11-44.

[4] *Communications of the ACM,* (1993) Special Issue on Augmented Environments, 36 (7).

[5] Edwards, W.K. (1995) *Coordination Infrastructure in Collaborative Systems.* Unpublished doctoral dissertation. Georgia Institute of Technology.

[6] Gaver, W., Smith, R. and O'Shea, T. (1991), Effective Sound in Complex Systems: The ARKola Simulation. *Proc. CHI'91,* pp 85-90, ACM Press.

[7] Hudson, Scott E. and Smith, Ian, (1996) Electronic Mail Previews Using Non-Speech Audio, *CHI '96 Conference Companion,* ACM, pp. 237-238.

[8] Ishii, H. and Ullmer, B., (1997) "Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms," in Proceedings of CHI'97, ACM, March 1997.

[9] Pederson, E.R. and Sokoloer, T. (1997), AROMA: Abstract Representation of Presence Supporting Mutual Awareness. *Proc. CHI'97,* pp 51-58, ACM Press.

[10] Want, R., Hopper, A., Falcao, V. and Gibbons, J., (1992) The Active Badge Location System, *ACM Transactions on Information Systems.* Vol. 10 (1), pp. 91-102.

[11] Weiser, M. (1991) The Computer of the 21st Century. Scientific American 265(3):94-104.