The Application of Progressive Turn Taking in the Design of an Ambient Social Notification Appliance

by

Mark Erik Altosaar

A thesis submitted to the
School of Computing
in conformity with the requirements for
the degree of Master of Science

Queen's University
Kingston, Ontario, Canada
September 2006

Your file Votre référence
ISBN: 978-0-494-18687-9
Our file Notre référence
ISBN: 978-0-494-18687-9

Abstract

Today, people are surrounded by many information providing devices, such as cellphones and BlackBerrys. These devices can be distracting when they are providing notifications that are not of immediate interest to the user. In this thesis, we present AuraOrb, an ambient notification appliance that deploys progressive turn taking techniques, so as to minimize such disruptions. AuraOrb uses eye contact sensing to detect user interest in an initially ambient light notification. When eye contact is detected, it displays a text message with a notification heading visible from 360 degrees. Touching the orb causes the associated message to be displayed on the user's computer screen. When user interest is lost, AuraOrb automatically reverts back to its idle state.

This thesis discusses observations made from previous work in ambient displays and sensing systems that we consider important in designing less disruptive devices. As well, we present three design principles which guided the decisions that led to the development of AuraOrb.

We performed an initial heuristic evaluation of AuraOrb's functionality using a set of heuristics specifically tailored to ambient displays. We compared progressive notification with the use of persistent ticker tape notifications and Outlook Express system tray messages for notifying the user of incoming emails. Results favored progressive notification over a continuous ticker tape display.

Acknowledgments

I would like to take this opportunity to thank my parents Erik and Leena. Words cannot describe how appreciative I am of their support over the years. Without their love and encouragement, I would not have been able to accomplish all that I have. I'd also like to thank Cheryl for her support (especially all the great meals) over the past two years.

I would also like to thank my supervisor Roel and all my friends from the Human Medial Lab. The experiences that I have had there can be considered anything but dull.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis describes our work in Attentive User Interfaces (AUIs), an area of Human Computer Interaction (HCI). It presents observations and design principles for the development of appliances that provide information to the user in a non-disruptive manner. These principles are implemented in a device called AuraOrb, which senses user attention to make intelligent decisions regarding information display. The goal of this work is to enhance the interaction of users with the many information-providing devices that are in their environment.

1.1 Motivation

We live in a world of ubiquitously connected computing devices. One of the problems when surrounded by all this technology is that users are all too often distracted by incoming notifications of emails, phone calls and system status messages [9].

Although computing devices are now ubiquitous, most of these devices are not designed to be unobtrusive. This is because computers are binary machines, making it difficult for them to move fluently from an ambient to foreground state and back. Moreover, most computers are ill-equipped to detect user interest in their communications. Ideally, the devices would know their local context of embedding: where they are, who is using them and for what purpose.

1.2 Objective

When computers were first introduced to the workplace, it was not uncommon for each worker to have only one personal computer (PC), and user interfaces were designed for this one-to-one interaction. Even though users now have a one-to-many relationship with many computers and devices, dialogue design is still based on a one-to-one relationship between the user and computer. Consequently, each device tries to dominate user attention without taking into account the other demands for attention from other devices, tasks or people. This behavior leads to frequent inappropriate notifications that may disrupt the primary task focus of the user.

The goal of this thesis is to design a computing device that integrates into the user's surroundings. As well, it should be able to provide timely notifications in a non-disruptive manner that reduce demands on the user's attention when interacting in a multiple-computer environment.

1.3 Contributions

This thesis contributes to the field of Human-Computer Interaction in a number of ways. First, we present four observations for progressive notification design, derived from an examination of previous work in ambient devices, notification systems, and studies in human group conversation. As well, three design principles for progressive turn taking notification devices are presented. Informed by the observations identified earlier, we created the design principles as a guide for our ambient attentive notification appliance, AuraOrb.

The main contribution is in the design and implementation of AuraOrb, an attentive notification appliance. AuraOrb uses eye contact sensing to determine user interest in its initially ambient light notification, before progressing its notification strategy. AuraOrb is our solution to the problem of inappropriate notifications that disrupt the user.

We performed a heuristic evaluation of AuraOrb where its progressive notification strategy was compared to ticker-tape notification as well as an email task-bar notification. The results favored the progressive notification design. Users commented that the use of eye contact sensing provided them an easy way to access more information about notifications.

1.4 Outline of Thesis

This thesis will be presented in six chapters. In the second chapter we will provide an overview of the field of HCI, and describe the paradigm of Attentive User Interfaces. In addition, previous work in ambient and peripheral displays will be provided. Chapter Three will discuss progressive turn taking and a number of observations and design principles that may be considered useful in the design of progressive notification appliances. In Chapter Four, we will present and discuss the design and implementation of our progressive turn taking notification appliance, AuraOrb. Chapter Five will include results from a heuristic evaluation in which different notification strategies were compared. Chapter Six concludes by summarizing the research and suggests directions for future research in the field of ambient notification and progressive turn taking appliances.

Chapter 2

Background

In this chapter, we will first provide a brief overview of Human-Computer Interaction and how it led to the development of Ubiquitous Computing and Attentive User Interfaces (AUIs). We then present related work in the areas of: notification systems, sensing systems, ambient displays and peripheral displays.

2.1 Human Computer Interaction

The scientific study of computers began in the 1920s, before the digital computers we are familiar with had even been invented. One goal was to develop computing machines that automate the work of human clerks, who would compute instructions sequentially [42]. Early user interfaces with computers were not interactive, and involved complex commands, input manually with knobs and switches, or via punch cards. Early computers were large and expensive, and required highly trained personnel. One computer would be used by many people, and thus, its interface was designed for that.

As computer technologies, and their use increased, interest was generated in the way in which humans use computers. This led to the creation of the Ergonomics Research Society in 1949, whose goal was to study the physical characteristics of machines and systems and how they impact the performance of the user. [7]. As work at the Society progressed, its researchers began to also consider physiological and psychological aspects of human-computers interactions, and how these could account for how humans used computers. This was an important step as it opened computer science research to new directions.

It is interesting to note that the initial goal in developing computers was to replace the human clerk, but the aim soon became to extend the ability of humans through the use of computing power. In the article "As We May Think" written in 1945 by Vannevar Bush [3], the author discussed how the recent advances in electronic machinery and computing should be used to augment and aid human ability to process data. He proposed the *Memex*, a storage system using electromechanical controls and microfilm cameras for quick retrieval of documents, books, records and communications. Bush envisioned the storage system consisting of microfilm frames filed with code numbers. He also designed for adding information into the system through the use of a photographic mechanism that would allow the user to input hand-written notes, pictures and letters.

The key aspect was that the documents be linked and indexed for easy access and for display on a desktop viewing panel. Bush realized that to aid humans in their data processing capabilities, systems should be designed around how humans handle data. Traditionally, information would be sorted either numerically or alphabetically, whereas the human mind searches via association. This led to the linking index of the Memex. Using the coded numbers on the microfilm storage, links were formed between disjoint frames. The sequence of links that emerged was known as a trail, and was the primary means of accessing data.

In the decades that followed, there was much research done into how computers might extend the human cognitive ability. J.C.R. Licklider presented the idea of *Man-Computer Symbiosis*, which was influenced by Bush's views on the computer's ability to augment human data processing capability [17]. Licklider proposed that the rapid increase in computing power be combined with the cognitive reasoning skills of humans. He described this *symbiosis* as the successful interaction of humans with computers where "making decisions and controlling complex situations" would be a cooperative process. Leveraging the data processing abilities of computers would enable the user to instead focus on "insight and scientific pursuit", the real source of innovation.

Douglas Engelbart, also inspired by the ideas of Vannevar Bush, studied how humans could interact with computers, and was especially interested in the features that allow for efficient information production. Engelbart designed the Graphical User Interface (GUI) to allow the user to interact visually with many different types of information simultaneously. To facilitate easy navigation in his GUI, Engelbart developed the computer mouse seen in Figure 2.1 which extended human motor ability. The GUI as well as the mouse were presented on December 9th, 1968 in San Francisco [27]. During this presentation, Engelbart also introduced the concepts of hypertext, object addressing, and dynamic file linking, as well as realtime networked collaboration with audio and video, where each user was able to share the contents on screen. The system he presented not only helped to fulfill Bush's idea of the Memex, but it also laid the foundations for modern day computing.

Figure 2.1: Douglas Engelbart's prototype mouse design [27].

2.2 Ubiquitous Computing

After Engelbart's 1968 presentation, researchers began experimenting with graphical user interfaces. In these interfaces, a combination of a mouse and a keyboard were used to navigate around a virtual desktop that had Windows, Icons, Menus and a Pointer (WIMP). These interfaces allowed the user to access more information simultaneously through the use of overlapping windows and dialog boxes. As the development of graphical user interfaces progressed, they became more complex as more and more information was vying for the user's attention. This led to complicated interfaces that potentially interfered with the actual task the user was trying to perform.

Mark Weiser, a researcher at Xerox's Palo Alto Research Center, realized that computers and their *interfaces* were becoming the focus of attention. Ideally the interface should be transparent, allowing the user to use the computer as a tool [40]. To address this, Weiser introduced the concept of Ubiquitous Computing. Ubiquitous Computers are essentially invisible devices, only providing the necessary functions required to perform the task. Rather than having one computer do many tasks, he proposed that we have a large number of these specialized devices in our environment, and that they be wirelessly networked and have sensing capabilities. Each device should be aware of their location as well as the presence and identity of the people interacting with the device. This knowledge of people and location would allow the computer to provide information that is context-aware and relevant to the user.

One of the key design aspects of Ubiquitous Computers, as introduced by Weiser [40], is the notion of Calm Computing [41]. According to Weiser, Calm Computers provide information without distracting a user from a primary task. One of the ways Calm Computers achieve this is through the use of peripheral display channels, channels that allow users to absorb information in parallel to their primary task. The use of calm technology should allow users to digest a greater amount of information from many different devices, without increasing the mental cost of interacting with each device; each device would be specifically designed to display its information without disrupting the user while interacting with other devices. With multiple devices surrounding the user, they are no longer tied to a single device (i.e. desktop computer) for interaction. With Calm Computing, the focus of computing moves from a one-to-one relationship to a one-to-many situation.

2.3 Attentive User Interfaces

To realize the vision of Calm Computing, where a network of computers surround the user providing information, an interface for interacting with these devices is required. Vertegaal et al. [36] have proposed a solution for interacting with groups of computers in a social manner, the paradigm of Attentive User Interfaces (AUIs). Rather than modeling the notification behavior of devices on a one-to-one relationship with the user, AUI proposes that we use the turn taking techniques employed by humans in conversation as a design metaphor for negotiated interruption by notification devices. The goal of AUIs is an optimal distribution of user attention across multiple tasks.

2.3.1 Properties of Attentive User Interfaces

To achieve this optimal distribution of attention, it is thought that these five properties of AUIs allow a device to provide information in a non-disruptive manner [36].

1. *Sensing attention*: By monitoring the user's physical proximity, body orientation, and eye fixations, AUIs can determine what device, person, or task the user is currently attending to in real time.

2. *Reasoning about attention*: By modeling user attention, AUIs can estimate task prioritization and predict attentive focus.

3. *Graceful negotiation of turns and sensing user acknowledgment of the request*: Before taking the foreground, AUIs determine whether the user is available for

interruption given the priority of the request, signal the user via a nonintrusive peripheral channel, and sense user acknowledgment of the request.

4. *Communicating attention*: To encourage efficient turn taking, AUIs communicate their attention to users, and communicate the attentive focus of the user to other AUIs and remote people that request the user's attention.

5. *Augment attentive resources*: Analogous to the cocktail party effect (the ability to focus one's listening attention on a single talker among a mixture of conversations and background noises), AUIs may optimize the use of the user's attentive resources by magnifying information in the estimated focus of the user activity while attenuating peripheral detail.

2.3.2 Human Group Behavior

The principles of AUIs arose from studying how humans behave in group conversations. In multi-party conversations, humans use peripheral nonverbal cues such as eye gaze and para-linguistics to decide when they should speak or be silent [37]. The ensuing turn taking process ensures that each listener can dedicate their full attention to the current speaker. Eye contact is one of the chief nonverbal cues through which humans manage conversation. As a nonverbal visual signal, it negotiates attention without interfering with the primary communication task. Research has shown that in four-person conversations, the looking behavior of an individual predicts with about 80% accuracy to whom that individual is listening or speaking [37]. For example, when someone is deciding if they should take the floor, they look to see if other people are making eye contact with them, indicating that they are waiting for that person to take the initiative and begin speaking. When that person is finished speaking, she will most likely look at the person who she wants to respond. This turn-taking process ensures that the conversation flows smoothly between listener and speaker, simply through the use of eye gaze.

2.3.3 Communicating with Devices

Being able to negotiate interactions in ubiquitous environments requires a communication channel that provides accurate indication of the user's interest. Using eye gaze as an indicator of interest was examined in the Look-to-Talk (L2T) interface [30]. In the L2T interface, the user's eye gaze opens a channel of communication with automated computer agents. A similar concept was used in the development of AuraLamp [19], seen in Figure 2.2. AuraLamp is a social appliance that uses eye-contact sensing to open up a channel of communication. It demonstrates how contextual speech recognition can be achieved through the use of eye contact sensing. Only when the user looks at AuraLamp does it accept verbal commands. The sensing of eye gaze, such as with L2T and AuraLamp can address the problem of addressing many devices in a ubiquitous environment.

AuraLamp belongs to a group of household objects augmented with eye-contact sensors called EyePliances [32]. The devices are all connected via either wireless or wired networking and share attentional information about the user. It has been shown by Oh et al. [30] that people look at a device to which they are speaking. Because EyePliances have knowledge of which device the user is currently looking at, only the device being attended to needs to listen to voice commands. This provides an intuitive interaction scenario for the user. Programming of these devices is also made simpler. Using eye contact as the indicator to start listening, the device needs to only be programmed to accept commands that are relevant, rather than having to recognize all speech and having to filter out unrelated commands.

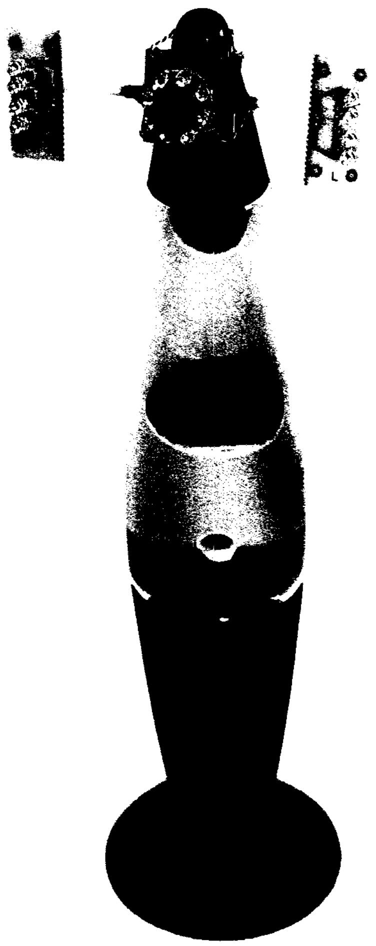

Figure 2.2: AuraLamp with Eye Contact Sensor.

2.4 Notification Systems

Notification systems communicate "additional digital information from sources secondary to current activities" [24]. A challenge facing notification system design is providing information in a manner that does not disrupt the user's attention. To address this, McCrickard et al. presented the following criteria for notification systems [23]:

- Notification systems should allow access to information in an efficient manner, and without interruption of the user's primary task.
- When notification systems are deployed in the environment, they should use ambient methods to communicate information. This ambient behavior can be achieved by subtly altering elements of the user's environment to convey information for background processing.
- Notification systems should support the user in allocating his attention between tasks, and once attention has been received by a device, allow access to additional information.

McCrickard et al. identified three modes of interaction with notification systems: interruption, reaction, and comprehension. Ideally, the interruption is minimal so that the user does not lose focus on their current task. McCrickard pointed out that in cases where the notification is important and where a fast reaction is required, the interruption can be more explicit. As well, in cases where an important notification has to be made, the information should be presented in a manner such that it is easily comprehended.

Similarly, Matthews et al. [21] identify three key issues in the effective operation of notification devices:

- Choosing the appropriate level of abstraction for displayed information.

- Choosing the appropriate level of notification.
- Using information transitions to signal requests.

The proper level of abstraction of data is important, especially when the communication channel is limited in bandwidth. This may be the case in an Ubiquitous Computing situation, where the device may be highly specific to a single task. Choosing correct notification levels for incoming events allows devices to mediate user interruptions. Finally, it is in the display transition that the user is made aware of the information.

Horvitz et al. [10] presented the Priorities notification classification system. The Priorities system uses learned user behaviors and explicit user feedback to automatically assign an urgency code to new notifications. In their paper, they discussed its application to the filtering of incoming email to help manage the user's attention when faced with a large number of messages.

2.5 Sensing Systems

In order to reduce interruptions in a scenario where many computing devices are present, it may be helpful to sense user interest. For effective multiparty communication between devices and a user, these devices must also be able to sense their surroundings and communicate their negotiations effectively.

Bellotti et al. presented a framework for the design of sensing systems [2]. They stressed that when designing these systems, the communication aspects of interaction should be the focus. They identified five questions that are relevant for designers of devices that are deployed in the user's environment. The questions they posed are specifically targeted to expose challenges that do not affect traditional graphical user interface interactions.

1. *Address*: How do I address one (or more) of many possible devices?

2. *Attention*: How do I know the system is ready and attending to my actions?

3. *Action*: How do I effect a meaningful action, control its extent and possibly specify a target or targets for my action?

4. *Alignment*: How do I know the system is doing (has done) the right thing?

5. *Accident*: How do I avoid mistakes?

Along with these questions, Bellotti et al. presented example systems that used sensing of human attention or behavior and sensing of other devices in proximity to solve some of the problems. For example, Augmented Objects [39] are devices that use RFID (radio frequency identity) tags and IR (infrared) sensors and emitters to determine which device the user is interacting with. They stressed that as devices become more embedded in the environment, user input has to be obtained through other means, rather than the standard keyboard and mouse. As long as objects are distinctive and suggestive of their action, the range of possible actions may be known. For example, a pair of scissors is an object whose action is suggested by its shape and the range of its possible actions is clear - using them to cut paper. As well, traditional responses that users receive upon actions in the GUI (e.g. dialog box or bouncing icon) may not apply when dealing with devices with no or limited text output capability. Instead, these devices can use visual and auditory channels to provide feedback when the user is interacting with the device.

2.6 Ambient Displays

One of the first papers to address the design of Calm Technologies was ambientRoom [16]. One of the important design criteria for ambientRoom was to allow users to absorb information parallel to a focus task by deploying visual and auditory channels low in symbolic content. The room was designed as an interface to information for processing in the background of awareness. This is in contrast to GUIs, which require constant foreground visual attention. One of the design goals of ambientRoom was to transfer information to different media. For example, the lighting system in the ambientRoom was devised to simulate the sun's pattern of illumination, so as to subtly indicate time of day. A wall-mounted clock in the ambientRoom allowed users to verify the exact time of day through a foreground symbolic display. The subtle alteration of the room's illumination is a good example of how using low symbolic displays allow users to monitor approximate quantities. The information sources were designed in such a way that the boundary between background and foreground awareness was fuzzy, allowing the user to absorb information in a more natural manner. Therefore, when a new process catches the user's attention, it can be smoothly integrated into their foreground awareness.

Another ambient display is Ambient Orb [6], seen in Figure 2.3, a commercially available ambient notification device. The orb uses gradual changes in color to notify the user of information. The orb can be configured to display information such as stock quotes, which allows the user to be aware of abstract values of the stock market. As a ubiquitous device with an aesthetically pleasing design, it blends well with the architectural surroundings of the user. Unfortunately, the limited information output of the display prevents any foreground interactions with the device.

Figure 2.3: Ambient Devices - Ambient Orb [6].

There has been relatively little work exploring the large design space that lies between ambient and foreground information display. One exploration of this space was LumiTouch [4], a pair of picture frames used to indicate remote presence. Through the use of proximity and touch sensors as well as LEDs that indicate status, the frames were able to transition between ambient and foreground states depending on the level of interaction with the device. The frames, seen in Figure 2.4, initially begin in an idle state, behaving exactly like regular picture frames. When one user is located near the display, sensed via a proximity sensor, the other picture frame begins to glow. This allows the users to be aware of each other's presence, in a ambient manner. Users can also progress to a foreground interaction by picking up the display and squeezing. This causes LEDs located on their partner's display to light up and indicate that the user is in direct contact with the display. The users can squeeze their respective displays to send simple messages back and forth along the lines of morse code.

Figure 2.4: LumiTouch ambient remote presence indicating picture frame [4].

Another example of a device that bridges the dichotomy between ambient and foreground display is the CareNet Display [5]. The CareNet display was inspired by the Digital Family Portrait [28], a picture frame that provides a small amount of information about the person in the portrait through a glowing frame. There are multiple frames, with each one covering a specific aspect, for example, health, location and mood. The CareNet system extends this idea and uses a digital picture frame to convey the status of an elderly person to a caretaker, where icons painted around a picture of the person indicate the status of the person on a number of health and other parameters. The icons allow the caregiver to be subtly aware of the status of the person under care in the periphery of attention. Touch sensors integrated into the display allow caretakers to examine information in more detail through foreground interactions. The picture frame is located in the home of the caregiver, and allows them to be aware of the person under their care.

Another exploration into the space between ambient and foreground displays and the associated interaction techniques was presented by Vogel and Balakrishnan in 2004 [38]. In their paper, they developed design principles and an interaction framework for public ambient displays that also allow transitions between implicit and explicit interaction techniques. The challenge they faced was to provide information awareness through ambient channels for multiple users, yet allow foreground interactions for a single user accessing private information. To achieve this, they identified four levels of interaction and display for a public computer controlled large plasma monitor:

1. Ambient Display.

2. Implicit Interaction.

3. Subtle Interaction.

4. Personal Interaction.

Initially, the display would be in the ambient state, and as the user located and oriented themselves towards the display, it would respond by providing a graphical representation of their monitored data. Moving closer to the display would increase the amount of information being displayed. At this point, the user can use a simple gesture set to control the information overview. To reach the personal interaction stage, the user can step closer to the display. They performed a user study and found that users commented that the use of body orientation and location was a natural way to transition between different abstractions of data. Vogel and Balakrishnan suggest that initial ambient display of information allows users to be aware of information without requiring foreground interactions. As well, sensing the user's proximity to the display allowed for a non-intrusive measure of interest.

2.7 Peripheral Displays

One shortcoming of ambient displays is that they cannot accommodate a large amount of information. Peripheral displays are a form of ambient displays that are able to provide more detail, but require the user to look at the display directly.

One such peripheral display is the Scope [34]. Scope displays multiple notifications using a radar-like visualization window, and the authors classify it as a "glance-able notification summarizer." Their goal was to summarize incoming notifications, and group them in an intuitive way so as to reduce information overload. Scope uses the previously discussed Priorities [10] system to automatically rank incoming notifications. Notifications are represented as small icons placed on the radar screen; the closer these icons appear to the center, the more urgent their notification. The arrival of new notifications is indicated by fading an icon onto the radar screen, and assigning it a color code. To help the user distinguish between icons belonging to different tasks, the authors split the radar into four color-coded quadrants: Alerts, Inbox, Tasks and Calendar.

The radar is initially displayed at the bottom right corner of the user's desktop. When the user moves their cursor onto the Scope, the window doubles in size, and moves to the foreground. The user can then select icons to explore detailed notification information. One of the shortcomings of this system is that the user must bring Scope to the foreground to find out more information, which obscures the task that the user was originally working on.

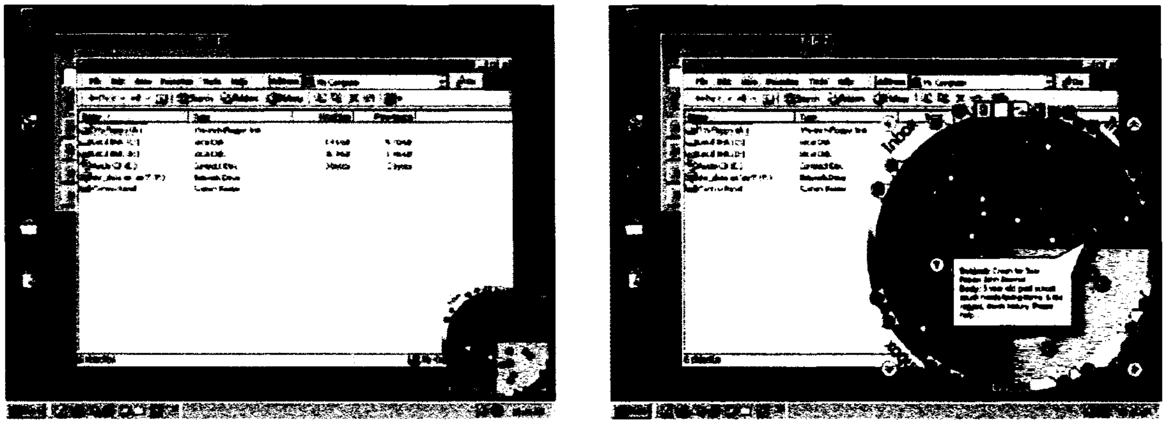

Figure 2.5: As the user mouses over the Scope notifier, it moves to the foreground of attention (Right-hand Image) [34].

This shortcoming is evident in Figure 2.5 where the notification radar is initially small. As the user mouses over Scope, it becomes larger and moves to the foreground of attention. Additionally, having a system that automatically adjusts the urgency ranking may cause the user to ignore notifications that are of higher interest because it is difficult to accurately predict user interest. Another problem that the designers of Scope encountered was the steep learning curve. Users were often not aware that new notifications had arrived and had difficulty differentiating between the icons.

Similar in design to the Scope, InfoCanvas is a peripheral information window able to present data from multiple sources of notification [26]. The authors of InfoCanvas contended that people have much information that they would like to be aware of (stocks, weather, traffic, emails, etc.) but the current methods for displaying this information require cognitive effort and increase distraction. The three key design goals, which are applicable to all aspects of information display, for the InfoCanvas were:

1. *Personalized and consolidated*: People should be able to choose information that is personally relevant, and the display of this information should be brought together to promote efficiency.

2. *Elegant and peripheral*: Displays of the information should blend smoothly into a person's home or work area and enhance a person's environment. They should not detract from other work spaces such as the person's computer desktop.

3. *Sensitive to privacy*: Displays should abstract or hide details of sensitive personal information such as financial accounts.

With InfoCanvas, users were able to monitor data through pictorial representations displayed in the periphery of the user's work area. The designers used an LCD display framed like a actual picture (seen in Figure 2.6), powered by a desktop computer. Another design goal for InfoCanvas was to convey information in an aesthetically pleasing way, thus reducing the obtrusiveness of associated notifications. For example, users can monitor incoming emails by having their friends' faces appear on the canvas. Initially, the user would notice a change in the display through their peripheral vision. However, one glance at the display would allow them to determine the sender of the email.

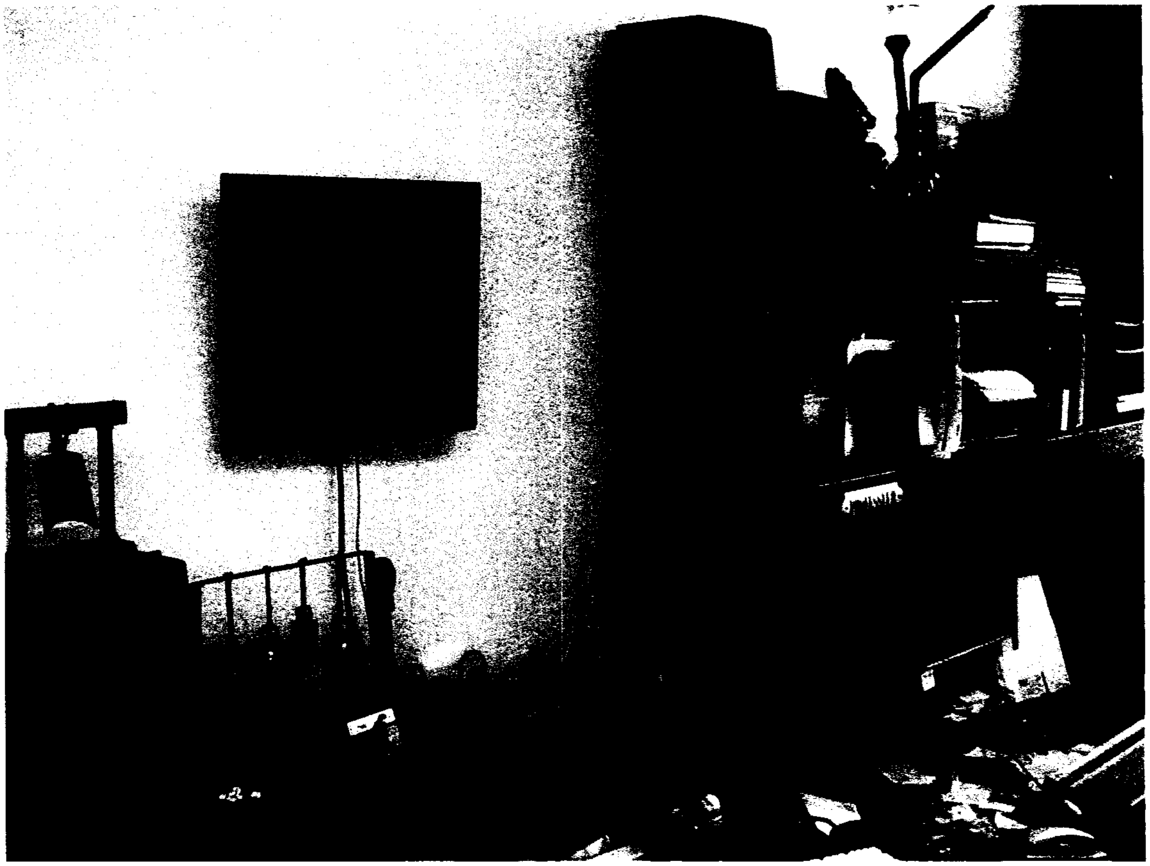

Figure 2.6: InfoCanvas peripheral digital notification art-piece [26].

In 2006, Hsieh, Wood and Sellen presented *Peripheral Display of Digital Handwritten Notes* [12]. In this system, they positioned a secondary monitor in the periphery of the user's workspace, for a low-distraction method of presenting information. The aim was to keep relevant information at the user's fingertips so as to aid the user in remembering information from many sources. They found that the most difficult part of design was the addition and removal of the notes on the display, as it would often distract users when items popped on or off the monitor. Additionally, as with other peripheral displays, foreground attention is required; in this case, a task switch to activate the program with the notes, is also needed to determine the exact information content of the notes.

2.8 Summary

In this chapter, we have provided a brief background of Human-Computer Interaction. One of the key concepts developed was that of Calm Computing. As we move towards a society where computing devices surround us on a daily basis, it is important that all these devices work together and provide information in a non-disruptive manner. Ambient displays are able to address Calm Computing, though some of the discussed systems lack sensing mechanisms to provide information-rich foreground interactions. The paradigm of Attentive User Interfaces suggests that we adopt the progressive turn taking behavior of human group conversation as a sociable means to move from background to foreground modes of interaction. In this thesis, we hope to design a device that is able to provide notifications, which can inherently be disruptive, in a calm manner through the use of progressive notification. In the next chapter, we present our observations on how these calm notifications may be achieved.

Chapter 3

Progressive Turn Taking

"Progressive turn taking" is the idea that devices should sense whether the user is actually interested in a notification before deciding to move into the foreground. Progressive turn taking is a primary means for achieving the goal of Attentive User Interfaces: an optimal distribution of user attention across multiple tasks. This chapter presents four observations that we have identified for the design of progressive turn taking notification devices. We also present three design principles that led the development of our ambient notification device; AuraOrb.

3.1 Observations for Progressive Notification

3.1.1 Observation 1: Visual Interest as a Measure for Cost of Interruption

The findings of McCrickard and Chewar [22] indicate that the choice of notification level very much depends on the action required upon notification. Notifications that require an immediate user response may require foreground alerts, while notifications that keep the user informed of general patterns in data should stay in the background. According to Horvitz [11], choosing a level of notification in such a way that it corresponds to the perceived urgency of a message may require a large amount of reasoning about the context and content of the message. Additionally, Adamczyk and Bailey [1] found that the timing of interruptions is important in reducing the disruptive effects notification delivery can have on task performance.

A key benefit of sensing user interest in a notification display is that it allows for an interactive measure of the cost of an interruption to the user.

3.1.2 Observation 2: The Eye as an Ambient Information Filter

It is interesting to note that in many ambient and peripheral displays, the looking behavior of users functions as a means to move from background to foreground, fluidly altering the resolution and abstraction level of the perceived information between eye movements. This is because the resolution of the retina is not uniformly distributed, with visual acuity dropping off considerably beyond 2 degrees from the visual axis. For example, when users are not looking directly at the wall clock in ambientRoom, it acts only as a placeholder for information. Its symbolic information is effectively rendered invisible. The retina's sensitivity to movement is inversely correlated with visual acuity, with peripheral vision being more sensitive to movement. This also means any moving or flickering display can be perceived as "annoying" when used as a source of ambient information.

By sensing when the user is looking for symbolic information (and when not), devices may move effortlessly between ambient and foreground modes of display.

3.1.3 Observation 3: Gaze Signals Communication

When multiple devices surround the user, deciding which device should take the floor and communicate with the user is a difficult problem. As discussed earlier, Bellotti et al. presented a framework for the design of sensing systems. They identified that a common problem when using multiple notification devices is the difficulty in determining which device the user is addressing [2]. By sensing directly when the user is attending to a notification device, we may improve coordination of notifications among multiple devices in a way that disambiguates the target of user attention. This behavior would mimic the use of eye gaze as a turn taking signal in human group conversation.

Eye gaze is one the few nonverbal cues that cross-culturally indicate interest.

3.1.4 Observation 4: Smooth Negotiation of Communication

Humans are able to negotiate their communications with great subtlety, through awareness of their surroundings and by picking up on social cues. The following scenario helps illustrate how non-verbal cues can be used to open channels of communication:

"Jeff is in his office, having a phone conversation with his wife. He leaves his office door open to signal to others he is available for communications. Alex walks up to Jeff's office to ask him an urgent question. Alex could interpret the open door as an invitation to interfere with the ongoing phone conversation. However, he has a number of progressive signals at his disposal to signal his request.

First, Alex positions himself such that Jeff can see him. Proximity and body movement peripherally indicate the urgency of his message without the use of verbal interruption. This allows Jeff to establish the urgency of Alex's message, and wait for a suitable moment in the phone conversation to grant the interruption. A brief glance by Jeff at Alex may signal unavailability. Sustained eye contact from Jeff serves as an invitation to Alex to take the floor. Should Jeff decide to ignore Alex's request for attention, Alex may choose to move out of Jeff's visual field, thus withdrawing his interruption request."

The subtlety of the above interaction pattern, where communication is regulated through non-verbal cues, is typically lost when interacting with computing devices.

3.2 AuraOrb Design Principles

We envisioned AuraOrb as a social appliance [9], an ambient notification device that uses information about user interest to determine its notification strategy. Informed by the above observations, we identified the following design principles that helped guide the development of AuraOrb:

3.2.1 Principle 1. Avoid Foreground Display

When a user is engaged with a foreground display conducting a primary task, that display should not be used to provide a visual notification if that notification does not warrant immediate action [23]. By providing notifications over an ambient channel in the periphery of user attention, the user is allowed to determine his or her own priorities: continue the main task without visual interference on the primary display, or attend to the notification request.

3.2.2 Principle 2. Sense User Interest Across Devices

To determine user interest in the information they provide, peripheral devices should sense when they are being attended to. By sharing this information among devices, disruptive notifications can be kept to a minimum. This concept was first employed in EyePliances [32], where appliances augmented with eye contact sensors shared information about which device the user is currently attending to. This phenomenon was modeled after how humans communicate their attention in social situations by looking at the person they are interested in communicating with.

This social behavior among devices is important, as Iqbal and Bailey determined that users are less annoyed when systems time their interruptions appropriately [15]. Eye contact is an excellent indicator for devices to detect when they should communicate with a user. By timing interruptions to occur when the user is not engaged with a particular task, we can ensure that they will not be distracted from an important focus task. For this to happen successfully, all devices must be able to sense user interest and share that information with each other.

3.2.3 Principle 3. Negotiate User Attention

As Fogarty et al. have shown, the use of sensors can support intelligent negotiation of interruptions, which can aid in avoiding task disruptions [8]. By moving into the foreground only when user interest is detected, AuraOrb fluently moves from background to foreground styles of display. To avoid disrupting the user, we also believe that devices should stay in the background as much as possible, even when this means the opportunity for notification may be lost.

Chapter 4

AuraOrb

In this chapter, we present AuraOrb, a social notification appliance that uses eye contact sensing to detect user interest in an initially ambient light notification. The design and interaction patterns of AuraOrb were guided by the four observations and three design principles presented in the previous chapter. To achieve our goal of social notification, we modeled how humans indicate attention in group conversation by progressively supplying more information to the user as the user provides more attention to AuraOrb.

We begin the chapter with an overview of AuraOrb describing the hardware implementation. We follow with a potential scenario of interaction, along with a discussion of the progressive notification levels. The software that controls AuraOrb's behavior is then presented.

The final section summarizes and presents an application scenario that illustrates how AuraOrb might be integrated into a network of Attentive User Interface devices.

4.1 Implementation

AuraOrb is a notification device that uses eye contact sensing to detect when a user attends to an initially ambient light notification. Upon receiving sustained eye contact, AuraOrb automatically progresses its notification strategy from ambient light to textual display of notification subject headings. If the user requires further information, a simple touch of the orb brings the complete notification information to the last display the user was attending.

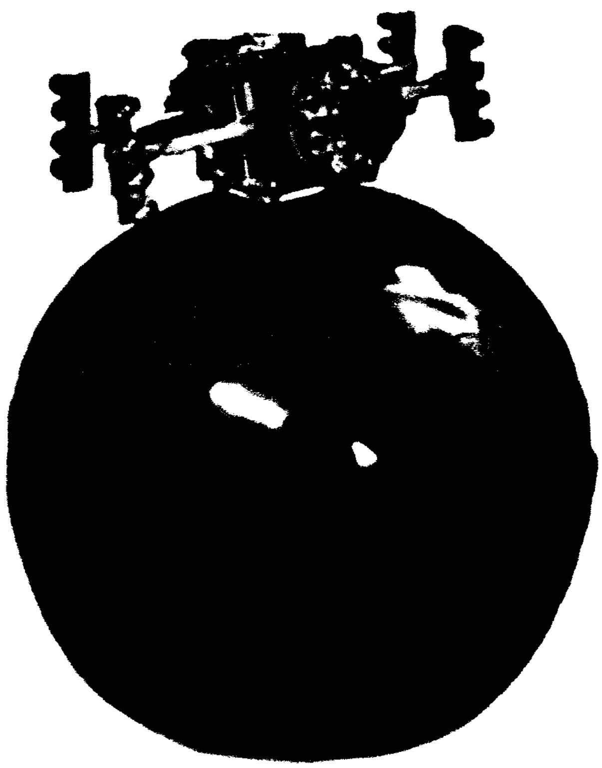

Figure 4.1: AuraOrb with Eye Contact Sensors.

AuraOrb, seen in Figure 4.1 was designed around a modified Olympia InfoGlobe OL3000 Caller ID device [14]. We augmented the Olympia InfoGlobe with eye contact sensors and touch sensors that allow AuraOrb to detect varying levels of user interest.

4.1.1 Hardware

Eye Contact Sensors

Mounted on the top of the AuraOrb are two eye contact sensors (ECS) [31] that point in opposite directions. The ECSes are capable of detecting when a user looks at the orb, and require no prior calibration of any kind. The ECSes have a range of approximately 50 degrees on either side of the display, and are capable of detecting eye contact from a distance of up to 1 meter. The AuraOrb's ECS is seen in Figure 4.2. Each ECS finds eyes through computer vision, using an infrared camera surrounded by a ring of on-axis LEDs. The on-axis LEDs mounted around the lens of the camera produce a bright pupil reflection in eyes within range, similar to the red-eye effect in photography. A dark pupil reflection is generated by a set of LEDs mounted off-axis to the lens.

Both sets of LEDs are flashed alternately in sync with the camera clock, producing bright and dark pupil reflections in alternate camera fields. Subtracting these fields allows a computer vision algorithm to identify the user's pupils. To determine when the user is looking at the ECS, the software looks for a corneal glint. When this glint is within the center of the pupil, eye contact is reported. The information is reported via TCP/IP to a connected server. The ECS software was previously developed at Queen's Human Media Lab, and was used unchanged for this thesis.

Visual Output

AuraOrb has two display modalities: an ambient glow and a ticker tape text display.

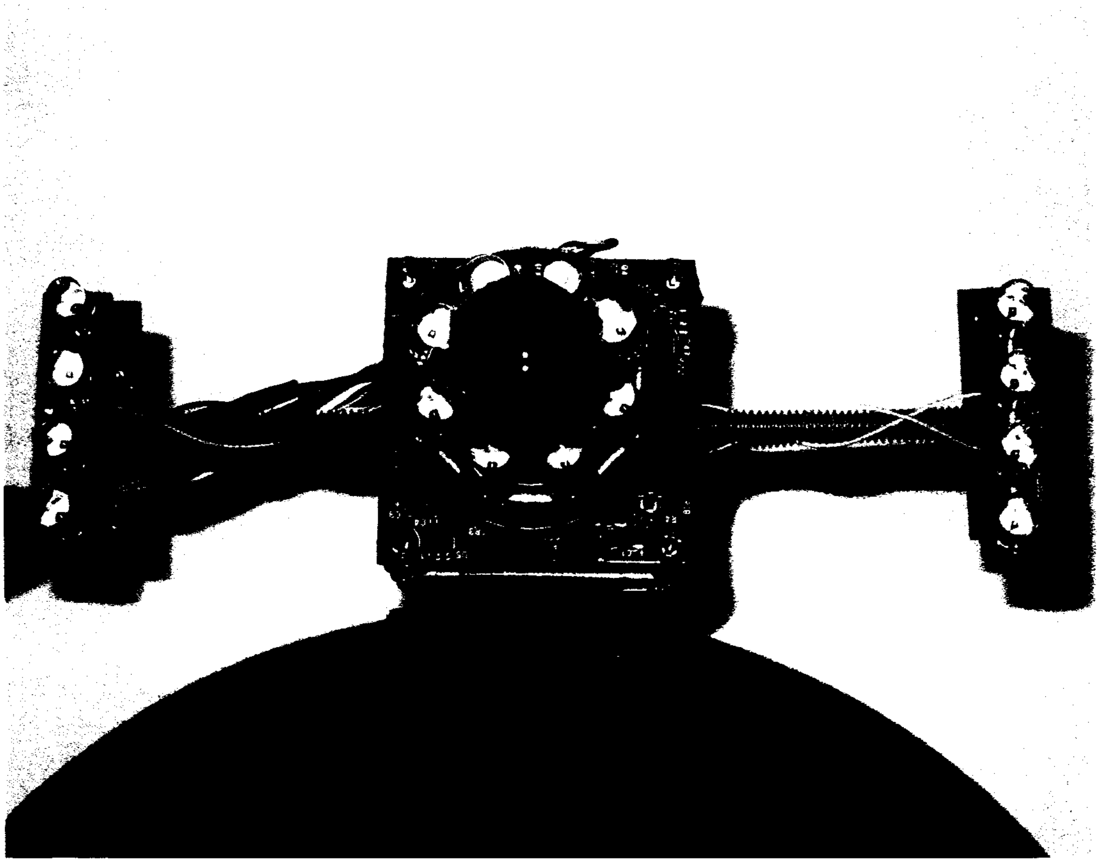

Figure 4.2: Eye contact sensors with on and off-axis LEDs.

The Olympia InfoGlobe provides a 36-character display visible from any angle. This display appears to float within the globe. This is achieved through the phenomenon of persistence of vision: a rapidly spinning disk within the globe paints the text using an array of 8 LEDs mounted on opposite ends. The text that is displayed is controlled through a caller ID interface that will be discussed later in this section.

To create the ambient glow, we coated the inside of the InfoGlobe with a UV reflective paint. This paint is transparent under normal lighting conditions, but glows when illuminated. Light is produced by a high voltage circular cold cathode UV light mounted on the inside of the globe. The ambient light is controlled via a Phidget [13] relay switch and a transformer to generate the necessary voltage. The relay switch controls the power to the transformer as the relay is only able to handle 100 V DC at 5 amps, and the power being outputted to the cold cathode is of low amperage, but very high voltage. The transformer accepts 12V DC, well within the operating range of the relay switch and outputs 680V DC to the cold cathode.

Phidgets are low-cost physical sensors and controllers. They are connected to a host computer via a Universal Serial Bus connection. The actual sensors are connected to a controller board (Phidget Interface Kit 8/8/8) that can handle both analog and digital inputs. The controller board is seen in Figure 4.3, where the relay board connects to one of the digital outputs, and the touch sensor (to be discussed in the next sub-section) connects to one of the the analog inputs. The benefit to using Phidgets is that they provide an easy-to-use programming interface.

Touch Sensor

To allow AuraOrb to detect foreground interactions, we also integrated a capacitive touch sensor Phidget in the body of the orb. To detect touch through the plastic globe, we secured strands of very thin copper wire along the inside perimeter. The capacitive touch sensor is normally able to detect touch through plastic, but we found that to increase sensitivity around the whole orb, the copper wire was necessary. This copper wire was soldered to the touch sensor board.

Caller ID Interface

We took advantage of the textual caller ID display built into the Olympia InfoGlobe to display the message headers. Because the Olympia InfoGlobe uses the MDMF

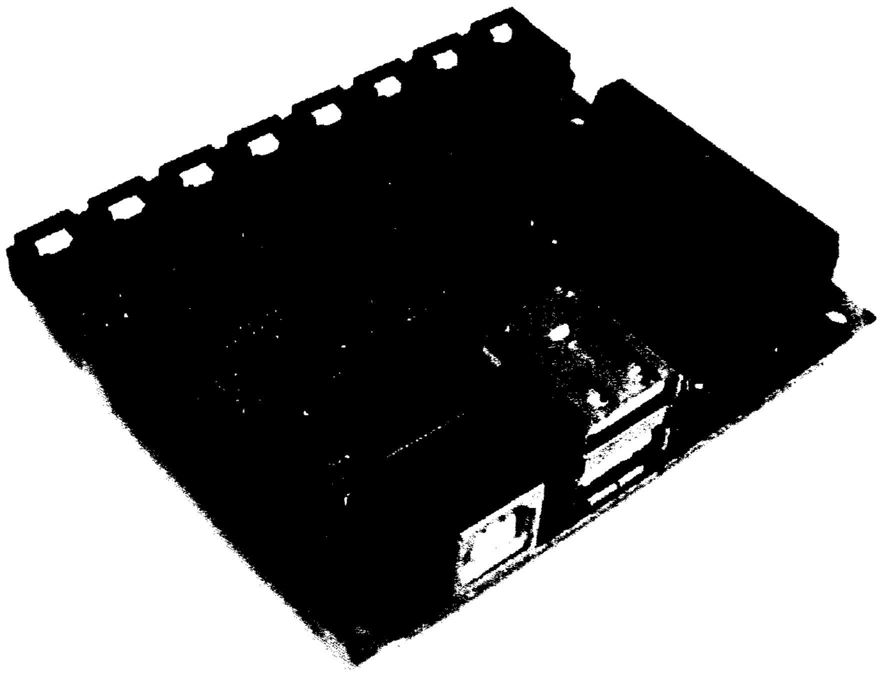

Figure 4.3: Phidget Interface Kit [13].

(Multiple Data Message Format) caller ID format, it allowed for both numeric and alphabetical characters to be sent over the phone line [25]. We used an adapter that interfaces the speaker output of the desktop computer sound card to the phone line input of the InfoGlobe. Messages were formatted for display on the globe using software that allowed us to generate caller ID format wave output [33]. The software solution will be discussed later in this section.

An interesting challenge that we had to overcome was disabling the demo mode of the InfoGlobe. After a period of inactivity, the InfoGlobe defaults to a demonstration operation mode where it shows the date and time on its ticker-tape display. Our initial solution used a Phidget relay switch to turn off the power leading to the display portion of the InfoGlobe (rotor mechanism and LED power supply). While this prevented the unit from going into its demo mode between notifications, we found that this introduced a delay when trying to display the message once eye-contact was detected. This delay was due to the motor having to spin the display mechanism up to the required speed to achieve the persistence of vision effect. We also experimented with controlling the LED power separately from the motor that drove the spinning disk. While this allowed us the ability to turn the ticker-tape display on or off at any point, we experienced erratic behavior of the LED output, with extra characters being displayed or with characters missing.

In the end, the solution to the problem was solved in software rather than hardware. We periodically sent blank Caller ID signals to AuraOrb to prevent it going into demo mode.

4.2 Interacting With AuraOrb

4.2.1 Scenario of Interaction

"Kyle is writing an article using a word processor on his desktop workstation. Kyle had previously used auditory notification for incoming emails but he disliked having to switch to his email program to see whether a message required action. He also finds visual notifications popping up on his primary display too distracting. Instead, Kyle has configured his AuraOrb to notify him of incoming messages through ambient light notifications. Figure 4.4 shows how Kyle's workstation display is augmented with an eye contact sensor. This allows Kyle's personal AuraOrb server to determine when he is engaged with his primary display.

Figure 4.4: AuraOrb showing ambient light notification.

When an email arrives, AuraOrb's software notices that the primary display is occupied and thus unavailable for visual notifications. Kyle's AuraOrb immediately lights up in bright blue, a color chosen by Kyle to indicate the arrival of email. Kyle sees the light appear in his peripheral vision but does not shift his attention to the orb until he has finished his paragraph. The AuraOrb maintains a steady glow in his peripheral vision. When he looks at the orb, it instantly displays the subject and sender of the email (Figure 4.5). Kyle decides the email can wait and shifts his attention back to his workstation. Noticing this shift, the software returns the orb to an idle state until Kyle receives a call. Kyle notices the red glow of the orb indicating an incoming phone call in his peripheral vision, and decides to look at the orb to see who the caller is. As he looks at the orb, it instantly displays caller ID information, showing it is his mother on the line. He decides not to take the call. As he turns to his computer screen, the orb returns to a steady red glow, turning idle only once the call is dropped. Another email arrives, and the orb glows a steady blue. When Kyle looks at the orb, he notices the email is from his mother. He decides to read it, and touches the orb. This causes AuraOrb's software to open a message window on Kyle's workstation screen and return the orb to an idle state."

Figure 4.5: AuraOrb displaying notification heading upon receiving eye contact.

The following scenario shows how a single AuraOrb can be used to notify the user of different types of incoming messages and notifications.

4.2.2 Progressive Notification Levels

AuraOrb is capable of three levels of notification:

- *Ambient Display*: The UV cold cathode lamp inside the globe allows the orb to glow with a color specific to the type of message. This form of display is purposefully non-symbolic by nature, and does not require focused attention to be understood.
- *Semi-Foreground Display*: The LED display provides a 36-character notification heading visible 360 degrees around the orb. Flicker produced by this type of display means textual notification should be avoided when the user is not watching the device. This form of display provides a symbolic summary of the content of the message.
- *Foreground Display*: Touching the orb causes the message corresponding to the notification heading to be displayed on the foreground screen last attended by the user. This modality requires a bitmapped, high-resolution screen to provide the full message content.

When a message arrives, AuraOrb's software first evaluates whether a notification should be posted on the user's primary screen. If the user is attending to their primary screen while performing reading or typing activity, the software subsequently illuminates the orb with a color specific to the type of message, as defined by the user (see Figure 4.4). When one of the eye contact sensors on the orb reports user interest in the notification, the software turns off the ambient notification light inside the orb. The orb's 360-degree ticker-tape displays the notification's heading. In the case of an email notification, this heading typically consists of the sender and subject of the message (see Figure 4.5). Upon touching the AuraOrb, the control software opens a window displaying the body of the associated message on the user's last attended computer screen.

Should the user decide to ignore an ambient notification, the software maintains the AuraOrb in its ambient state until it detects sustained eye contact from the user. When the user removes their gaze after having read the notification heading, indicating that they are not interested in finding further information regarding the notification, the orb returns to an idle state. This progression of steps by our control software illustrates how AuraOrb implements the three design principles for a social notification device:

- *Avoid Foreground Display*: With an ambient glow as the initial form of notification, we are able to move the notification alert to the periphery and off the primary display. By using this less symbolic method of notification, the user is made aware of the notification without requiring a task switch or foreground attention. They are able to attend to the notification when they deem it necessary.

- *Sense User Interest*: Through the use of eye contact sensing, both on the user's primary display and on AuraOrb, we are able to detect when the user has shifted their attention away from their monitor and to the AuraOrb.

- *Negotiate Attention*: Similar to a person entering a conversation, AuraOrb negotiates user attention through the use of sensing of peripheral cues such as eye contact and touch. AuraOrb only moves to the foreground of the user's attention when it knows that the user is actually paying attention to it. This avoids cases where AuraOrb may try to grab attention from other devices or people the user may be interacting with.

AuraOrb handles multiple incoming notifications in the following way. Upon the arrival of the first notification, the orb begins to glow. As more notifications arrive, the orb maintains its ambient glow. When the user makes eye contact with the orb, it immediately shows the number of messages that are queued. After a brief delay (1.5 seconds), the first notification header is shown. AuraOrb allows the user to step through notifications much like an answering machine. While the header of the notification is being shown, the user can double-tap the orb to show that notification on his primary screen. Conversely, if the user wants to skip the notification, all that is required is a single tap on the orb.

4.2.3 Software

The software that controls AuraOrb is discussed in this section. We begin with how we used the data that the Eye Contact Sensor system outputs. The next section discusses how the Phidgets input sensors and output relays are controlled in software. We conclude this section with an overview of the complete AuraOrb system and the components that communicate within it.

Eye Contact Sensing

The Eye Contact Sensors that we mounted on AuraOrb are connected to a Windows system that runs the glint detection algorithms. The eye contact information is output on a telnet port, which our control software communicates with via TCP/IP. The eye contact data is updated 30 times a second. In the first version of our software, the data being transmitted over the TCP stream was used directly. While this allowed AuraOrb to respond to eye-contact, we found that in some cases, if eye contact was lost for one or two frames, the system would assume that the user had looked away.

In the second version of AuraOrb's control software, we improved the eye contact data processing. A buffer was used to validate whether eye contact is reported. When 10 consecutive frames of eye contact are recorded, we set a boolean variable to true indicating that we had sustained eye contact. Once eye contact is sustained, the system would require 10 consecutive frames of lost eye before setting the boolean to false.

Phidget Interface

When the control software detected that the user has received a new notification (by checking the status of the user's email account), the software turned on AuraOrb's ambient glow. If the orb was already in its ambient state, and a new notification arrives, the software maintains the glow and adds the notification to a queue. The ambient light is controlled in software via a boolean variable that holds the state of the relay switch. The Phidgets API provided for simple implementation, as it handled the communication over the USB connection and to the Interface board to which the relay switch was connected.

Eye contact is reported in the system via a global flag that is connected to an event handler that looks for a change in that variable; when it is detected, the ambient glow is turned off. The system then monitors the Phidget touch sensor interface. This is done by checking a variable to which the Phidgets API provides access that contains the analog value of the capacitive sensor mapped from zero to 1000. We found that the sensor was very sensitive and a threshold variable had to be used. The value of the touch sensor variable would be updated each time the touch sensor was polled; thus, as was done with the eye contact sensor buffer, if the past 50 values that were greater than the threshold occurred within a short time period, we set a boolean touch variable to true. We detected the double tap behavior by checking to see if the boolean touch variable had a sequence of false-true-false behavior within a set period. We were able to adjust both the single touch and double touch times to suit individual users if we found that their taps on the orb were too fast or too slow.

Caller ID Interface

In the first version of the AuraOrb software, we used an API that allowed us to create caller ID formatted waveforms. Unfortunately, its behavior was not consistent, and we found that many messages were not being displayed, as the waveforms were not conforming to caller ID format.

The solution to this issue was to use a program written specifically for generating caller ID signals. There were a few programs available that produced the correct waveforms for telephone numbers, but we required a program that could output both character and number format signals, so we could send the email header information. The program we used is called CIDMage [33]. Using a program outside our programming environment introduced the problem of how to send the header information to the external CIDMage program.

A program to control the mouse and keyboard was written; it would move the mouse over the text input area on CIDMage, issue a click to activate the text box, and then by controlling the keyboard, the program would input the header. The mouse would then be moved to the 'Generate' button and clicked to submit the header text and the caller ID format wave signal would be generated by CIDMage. The one drawback of this approach is that it requires a separate PC executing just the mouse/keyboard program as it renders the computer unusable for any other tasks while our program is running.

4.3 System Overview

Three personal computers were required to operate AuraOrb.

1. *System One*: This computer was connected to the Eye Contact Sensors mounted on the AuraOrb. It performed the computer vision algorithms and operated a telnet server that distributed the eye contact status of the eyes visible to the cameras.

2. *System Two*: This computer was responsible for sending the caller ID formatted waveforms to AuraOrb via the CIDMage program. This required software control of the mouse and keyboard.

3. *System Three*: The final computer was running the AuraOrb control software (including the Phidget Interface), which communicates with the other two systems.

The following diagram (Figure 4.6) illustrates the flow of data between different components and computers in the AuraOrb system.

When AuraOrb's control software receives a notification of a new email, it connects to the ECS server connected to a user's computer monitor. If eye contact is reported

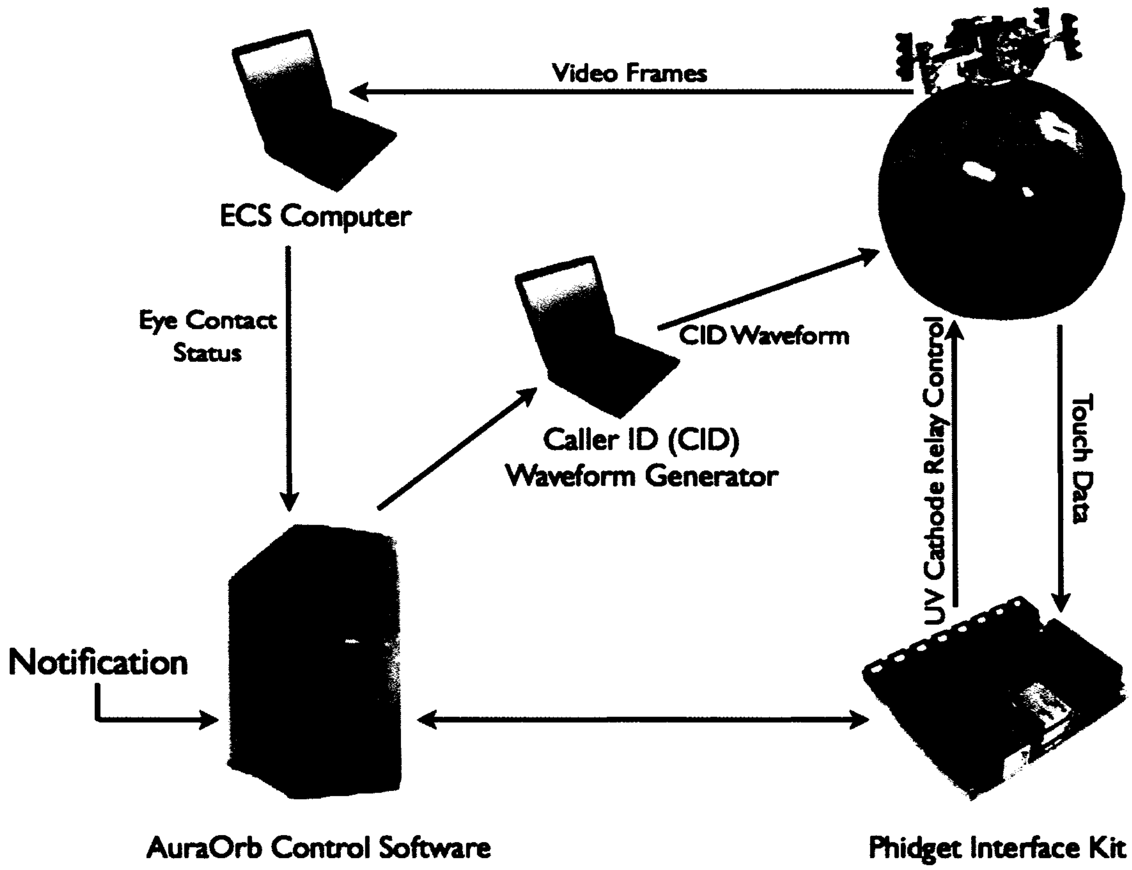

Figure 4.6: System overview of AuraOrb and its connected components.

for the display, the control software activates the Phidget relay and AuraOrb begins its ambient glow. The software then polls the AuraOrb ECS server to look for eye contact. When eye contact is reported from the ECS, the software also verifies that eye contact was lost from the user's primary display. If it is the case where the user has looked away from their screen and given their attention to AuraOrb, the software turns the Phidget relay off, and, through a socket interface, sends the message header information to the PC connected via audio output to the phone line input of AuraOrb. While the message is being displayed, the software continually reads the Phidget touch sensor variable looking for a single tap or double tap. The software is also checking to see if the user has looked away from AuraOrb by reading the values from its ECS server. When the software detects a double tap, it brings up Outlook Express on the user's computer and makes it the foreground window via a simple program running on the user's computer that communicates via TCP/IP.

4.3.1 Error Handling and Review Capabilities

We have designed AuraOrb to handle cases where there may be sensing errors or situations where the user would like to review notifications that may not have been attended to. Sensing errors may occur when eye contact is incorrectly reported. If the orb is in its ambient glow state and an eye contact sensing error occurs, the notification is queued internally by AuraOrb's software. To view the missed content, the user can look at AuraOrb and it will display the notification header of the last non-attended notification.

AuraOrb also supports situations where the user may have several notifications queued up. As previously mentioned, if AuraOrb is glowing and the user looks at it, AuraOrb displays the number of queued notifications. While browsing through the notifications, the user can double-tap to select a notification, which removes it from the queue. If at that point they decide to attend to the notification content on their primary display and look away from AuraOrb, it queues up the remaining notifications. After attending to the notification on their computer, the user can then look back at AuraOrb and it will once again display the remaining notifications.

4.4 Summary and Application Scenario

Through the use of eye contact and touch sensing, we were able to design and implement an ambient notification device that used the principles of Attentive User Interfaces, in particular progressive turn taking. The touch sensing capabilities and electrical control of AuraOrb were provided by Phidgets, and the eye contact sensor was developed in the Human Media Lab. Guided by the observations and design principles presented in the previous chapter, we described how AuraOrb's progressive notification strategy was implemented both in software and hardware. The following potential application scenario illustrates how AuraOrb can operate successfully in a network of attention sensing appliances.

In this potential scenario, the user is in their living room watching television. The television is an AttentiveTV, augmented with an eye contact sensor [35]; a television that watches the user. The user is able to move their attention away from the TV without missing any content. When the user looks away, the TV pauses, and when they look back, it resumes wherever they left off.

The user has positioned an AuraOrb in their periphery. This allows them to be made aware of incoming notifications without being distracting from their primary task, in this case, watching television. When a notification arrives, control software knows that they are watching TV as the Attentive TV reports eye contact with the user. AuraOrb's software turns on the ambient glow. The user is aware of the notification, but they can continue watching TV uninterrupted. A few minutes later, they decide to look at AuraOrb, The software sees that eye contact has been lost with the TV, and pauses the content. As soon as eye contact is reported by AuraOrb, the software sends the notification header of the email to the ticker-tape display on AuraOrb. In Figure 4.7, we see the user about to touch the orb as they have decided that they want to view the entire email. The tap is detected and the email is brought up on the last attended display (Figure 4.8). As the user looks away from AuraOrb back to the television, its ticker tape display is blanked, and it returns to its idle state. The user can now read the email on their TV, and, once finished, use a remote control to close the email and continue watching the television program.

Figure 4.7: User attending to AuraOrb while Attentive TV pauses television program.

Figure 4.8: User reading message on Attentive TV after selecting it via AuraOrb.

Chapter 5

User Evaluation

In this chapter, we discuss the results of our initial heuristic evaluation. Our findings are very encouraging and we believe that they validate the use of progressive turn taking in the design of notification appliances.

We begin with a description of the experiment and how we chose to evaluate AuraOrb. The results of a questionnaire are presented, and we conclude the chapter with some initial user experience comments.

5.1 Evaluation Details

To evaluate our design, we conducted an initial heuristic evaluation that compared the progressive notification techniques used in AuraOrb to a scrolling ticker tape notification, as well as to standard notifications provided by Microsoft Outlook Express on a personal computer platform. Our evaluation compared the following modes of operation:

1. *AuraOrb with progressive notification*: In this mode, AuraOrb operated according to our description in the previous chapter. When a notification arrives, AuraOrb begins to glow. The header of the notification is displayed upon user eye contact. To access the message associated with the notification, the user taps AuraOrb twice. This causes the message to be displayed on their monitor.

2. *AuraOrb with continuous ticker tape scroll*: In this mode, we evaluated the use of AuraOrb as a persistent ticker tape device placed in the periphery of the user's vision. Here, AuraOrb continually scrolls the number of unread notifications, much like the original InfoGlobe design (e.g., "2 New Messages"). When the user makes eye contact with the orb, the header of the first unread notification is displayed. The user further interacts with the orb as described in the first mode of operation, through physical contact of the orb.

3. *Outlook Express with system tray notification*: We evaluated the use of standard Outlook Express notifications as our baseline condition. We used the system tray icon for visual notifications of incoming messages, disabling auditory notifications. When new email arrives, a small icon appears in the system tray in the bottom right hand corner of the participant's screen. The participant either clicks on this icon or on the Outlook Express tab in the task bar to bring up the email program.

5.1.1 Participants

Eight participants, all male, were recruited from Queen's Human Media Lab for the heuristic evaluation. Ages ranged from 23 to 35. Three of the participants were usability experts. Two of the experts had 2 years experience each, and the third expert had over 8 years of experience. The remaining five evaluators were graduate students in Human Computer Interaction with previous experience in heuristic evaluation.

5.1.2 Heuristics

Heuristic evaluation is a low cost and informal way to identify usability problems in a product at nearly any point in the development cycle. The criteria identified by Nielsen are applicable to many areas of product design [29]. However, Mankoff et al. found that some of the heuristics were ill-suited to ambient display evaluation [20]; and created a modified set of heuristics tailored to ambient displays. They compared their set of heuristics to that of Nielsen's in a formal evaluation of ambient displays developed at UC Berkeley. Results demonstrated that evaluators were able to find more problems with ambient display designs with the modified set of heuristics. As well, their research showed that by using their modified set of heuristics, a small group of evaluators was able to find a large proportion of problems. The heuristics presented by Mankoff et al. were also applied in the evaluation of the peripheral information display CareNet Display [5]. The authors found that a group of 8 evaluators identified 75% of the known usability problems. In our evaluation of AuraOrb, we used the 12 modified heuristics listed in the paper by Mankoff et al.:

1. *Sufficient information design*: The display should be designed to convey "just enough" information. Too much information cramps the display, and too little makes the display cluttered and less useful.

2. *Consistent and intuitive mapping*: Ambient displays should add minimal cognitive load. Cognitive load may be higher when users must remember what states or changes occurred in the display menu. The display should be intuitive.

3. *Match between system and real world*: The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. It should follow real-world conversations, making information appear in a natural and logical order.

4. *Visibility of state*: An ambient display should make the states of the system noticeable. The transition from one state to another should be easily perceptible.

5. *Aesthetic and pleasing design*: The display should be pleasing when it is placed in its intended setting.

6. *Useful and relevant information*: The information should be useful and relevant to the users in the intended setting.

7. *Visibility of system status*: The system should always keep users informed about its current condition through appropriate feedback within reasonable time.

8. *User control and freedom*: Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialog. The system should provide undo and redo functionality.

9. *Easy transition to more in-depth information*: If the display offers multi-level information, the system should make it easy and quick for users to find out more detailed information.

10. *Peripherality of display*: The display should be unobtrusive and remain so unless it requires the user's attention. The user should be able to easily monitor the display.

11. *Error prevention*: Even better than informative error messages is a careful design which prevents a problem from occurring in the first place.

12. *Flexibility and efficiency of use*: The system should allow users to tailor frequent actions.

5.1.3 Procedure

For the purpose of the evaluation, a single AuraOrb was positioned within arm's reach beside a 17" screen displaying the Outlook Express email client. Prior to the test, the evaluator asked the participants to familiarize themselves with the heuristics. A brief demonstration of each modality of use with the device was provided by the evaluator. The evaluator then walked the participants through typical usage scenarios of the two Orb designs as well as the email client. In order to counterbalance any effect of practice, we used a Latin square to determine the order in which the conditions were provided to the evaluators. After the evaluation was completed, participants were asked to rank the severity of the usability problems found on a 5-point scale. They were then given a questionnaire with the following five open-ended questions:

1. Which notification method had the lowest cost of interruption?
2. Which method signaled the arrival of a notification most efficiently?
3. Which notification method allowed the fastest response?
4. Which notification method allowed for the most convenient access to more detailed information?
5. Which notification method did you prefer? Why?

5.1.4 Task

Each participant was allowed to spend as much time with each notification method as they felt necessary. To simulate a focus task, participants were asked to type the contents of an arbitrary newspaper clipping into a word processor presented on the main screen. The email client was not visible during this task, as it was minimized in the task bar. Participants were sent email notifications that interrupted their task. The emails were sent by the evaluator who was located at a computer terminal out of view from the participants. Emails consisted of messages relevant to the task, as well as simulated junk mail. Individual emails as well as batches of emails were sent to simulate a sequence of notifications, allowing users the opportunity to fully explore the use of AuraOrb as a device to handle typical incoming notification patterns.

5.2 Results

5.2.1 AuraOrb with Progressive Notification

For progressive notifications with AuraOrb, the majority of the usability problems identified were ranked not very severe and centered around obvious hardware limitations of the device. For example, participants noted that emails containing a long subject header were truncated on the orb display. While they were still able to use this information to decide whether to read the associated message, participants felt this violated the "sufficient information design" heuristic. We can attribute this problem to the limited number of characters supported by the InfoGlobe hardware.

Another problem was that upon the arrival of a notification, the Orb moved from its idle state to ambient light notification rather suddenly. This transition was distracting and violated the heuristic "peripherality of display." Once again, this was the direct result of a hardware limitation; our prototype UV cold cathode tube could not be dimmed to provide a smooth transition. Some users also commented that the level of illumination was too bright, which violates the "aesthetic and pleasing design" heuristic. Again, this focuses around a hardware limitation of being unable to vary the input voltage to the cold cathode.

5.2.2 AuraOrb with Continuous Ticker Tape Scroll

This method of notification was considered the most detrimental by all evaluators.

The two major problems identified were a lack of "Visibility of system status" and "Peripherality of display". Because the orb continually displayed the number of notifications on its ticker tape, users commented that they had to constantly monitor the orb to stay aware of the arrival of new information. This was because the display only updated the number of messages, that is, a single digit. Evaluators often noted that they felt like they would miss a notification if they did not repeatedly look at the orb.

Many evaluators reported attentional problems with the constantly scrolling ticker tape display. They commented that it was always grabbing their attention, and considerable cognitive effort was required to remain focused on the typing task.

Some comments recorded were:

"Ticker tape was the worst, it required constant monitoring to notice changes."

"I felt compelled to look at it regardless of pertinent information. There was this automatic attraction to the bright moving text in the corner of my eye."

These results coincide with previous evaluations of notification displays, where continuously updating peripheral displays were found to be more distracting and disruptive to a user's primary task [18].

5.2.3 Outlook Express

We also found problems with the Outlook Express system tray notification. Many users missed the appearance of the new mail icon in the system tray, violating the "Visibility of state" heuristic. This was rated as a very severe problem, as it caused most evaluators to miss the arrival of new emails.

Another usability problem was the difficulty that users had with acquiring more information about the notification heading. Participants did not want to switch from the primary word processing task to the email program to find out the subject heading.

5.3 General AuraOrb Comments

Some evaluators noted that they did not like that the email could not be displayed on the notification device itself. Additionally, when multiple notifications were queued up, evaluators did not enjoy having to go through each message in sequence. One benefit of using the Outlook Express inbox is that users are provided with an overview of all the new emails. Overall, however, participants enjoyed using AuraOrb with progressive notifications, as the eye contact sensing afforded the ability to display headings without having to switch tasks or use the mouse or keyboard. One user emphasized this when discussing the seamless transition between information states:

"I liked that information was there when I looked at it, as opposed to loading or switching to the email client."

When asked which notification method they would prefer, one user commented:

"I preferred AuraOrb's ambient notification because:

- It was clearly visible in the periphery.
- There was no need to periodically check it.
- Status change was very obvious."

Another user wrote:

"AuraOrb - many reasons:

- Cool, it would look great on my desk.
- I don't need to be situated near the computer, I can see if from across the room, just like the voice mail light on a phone."

Table 5.1 shows the results from the questionnaire. The results favored the AuraOrb with progressive notifications. It is interesting that when asked which notification method allowed for the fastest response to new notifications, half of the participants chose AuraOrb, while the other half chose Outlook Express. User comments indicated that in cases with a small number of emails, a quick glance at the orb provided the fastest response. As soon as the number of unanswered notifications increased, some evaluators preferred the parallel nature of the Outlook Express email client, which displayed all new emails after double-clicking the tray icon.

With regards to the convenience of accessing more detailed information, six of the eight evaluators preferred the progressive disclosure that AuraOrb's eye contact and touch sensors provided. They commented that they liked only having to bring

Table 5.1: Results from questionnaire. Cells indicate number of users preferring a device category.

	AuraOrb Progressive Notification	AuraOrb Tickertape	Outlook Express	AuraOrb + Outlook Express
Notification method with the lowest cost of interruption.	5	0	3	0
Most efficient signal of a notification arrival.	7	0	1	0
Which notification method allowed for fastest response?	4	0	4	0
Which notification method allowed for most convenient access to more detailed information?	6	0	2	0
Which notification method did you prefer?	5	0	1	2
TOTAL	27	0	11	2

up the email message on the primary display after reading the header on the orb, thereby knowing that the message would be relevant. Conversely, to access more detailed information with Outlook Express, the user must bring the email client to the foreground on the primary display, which may disrupt their primary task. However, one evaluator pointed out that if the new emails were important, the email client would already be open, and ready to display additional information.

In terms of which notification method users preferred overall, AuraOrb with progressive notification was selected by five of the eight evaluators while two chose a combination of AuraOrb and the Outlook Express tray icon and one chose the Outlook Express tray icon solely.

The results of our initial evaluation are encouraging, as they suggest that users enjoyed using AuraOrb to handle their incoming notifications. The use of progressive turn taking allowed them to make informed decisions regarding task switches, thereby minimizing distraction from their primary tasks. In the next chapter, we will summarize the results of the evaluation of AuraOrb with respect to the four observations we identified for the development of a social ambient notification device. The next chapter also contains the conclusion of this work presented in this thesis.

Chapter 6

Discussion and Conclusions

In this chapter, we present a summary of our work and conclude this thesis, as well as possible limitations and suggestions for further research in progressive notification appliances. We begin by examining the results of the initial evaluation in terms of the observations we identified earlier in this thesis.

6.1 Discussion

Results of our evaluation generally suggest that progressive turn taking techniques allowed AuraOrb users to access notification headings with minimal impact on their focus task. This section discusses some of the benefits and shortfalls of the evaluated interfaces, using our observations as a guideline.

6.1.1 Observation 1: Visual Interest as a Measure for Cost of Interruption

While our evaluations were not focused on establishing the actual cost of interruption, most evaluators were positive about the use of eye contact sensing as a lightweight means for establishing the importance of a notification. Eye contact sensing reduced the cost of interruption by removing the need for manual actions to disclose information about a notification heading. While the continuous ticker tape provided the same information upon eye contact, it increased the perceived total cost of interruption by being continuously active. As such, it cannot be considered Calm Technology [41]. As for Outlook Express, system tray notification icons required manual action by users to ascertain the importance of a message. This also increased the apparent cost of the interruption to the user.

6.1.2 Observation 2: The Eye as an Ambient Information Filter

Our results indicate that the use of a continuous ticker tape display for notifications may be detrimental to user focus [22] since this type of display attracts attention through movement in peripheral vision. This is particularly true for the kind of ticker tape display deployed in the InfoGlobe, which has an extremely low refresh rate. It appears the ticker tape in AuraOrb is useful only when the user is already looking at the device. Another issue with the ticker tape display is that, due to the limited resolution of peripheral vision, minor updates of the display may go unnoticed. The same problem occurred with the Outlook Express interface, most likely because the icons deployed for system tray notifications were too small. The use of eye contact sensing to detect when users are looking at its display gives AuraOrb the ability to adjust the resolution of its display according to that of the user's vision. One evaluator noted that the use of ambient light notification allows users to see notifications from a distance. At large distances, the resolution of the human retina does not allow for users to distinguish between textual notifications.

6.1.3 Observation 3: Gaze Signals Communication

Although the continuous ticker tape displayed notification headings upon eye contact, our evaluators did not perceive this transition as a change in the state of their communications with the device. AuraOrb's use of progressive notification upon eye contact seemed to fit well with evaluator expectations of the use of gaze as a signal for opening communications.

6.1.4 Observation 4: Smooth Negotiation of Communication

While AuraOrb's progressive notification strategy seemed to fit the idea of smooth negotiation of attention, as found in face-to-face communications, it did not handle multiple simultaneous notifications appropriately. This is because the sequential handling of multiple notifications opposes the process of progressively negotiating user attention. Users were required to skip through notifications manually to determine their appropriateness, making this interface worse than the parallel views of message headings provided by Outlook Express. This drawback and some possible solutions will be discussed further in the Future Work section.

6.2 Summary of Contributions

- We presented AuraOrb, our solution to the problem of disruptive notifications. AuraOrb uses sensing of the user's attention to determine the appropriate channel for notification. Its development was guided by our observations on related work and three design principles. We have designed AuraOrb to be a social appliance, and much like a considerate person in group conversation, AuraOrb is able to fit into a network of attentive devices and reduce the occurrence of disruptive notifications.

- We performed a heuristic evaluation where AuraOrb's progressive notification technique was compared to a ticker tape notification as well as a traditional desktop task-bar notification. The results favored AuraOrb's progressive notification design, with users enjoying the access to more information upon eye contact, and the ability to make informed decisions regarding task switches when deciding to accept notifications.

- We presented four observations for the development of ambient and progressive turn taking display devices. The observations illustrate that eye contact is a strong indicator of a person's attention. We believe that interacting with devices should follow the turn taking behavior employed by humans in group conversation.

- We established three design principles to guide development of social notification appliances. As a social appliance, an ambient notification device should use information about user interest to determine its notification strategy. The design principles identify design considerations that allow disruptions to be minimized.

6.3 Limitations and Future Research Directions

While the results of the initial evaluation are encouraging, there are limitations in the evaluation.

When performing heuristic evaluations, it is required that the evaluators are experts, or at the very minimum have previous experience in heuristic evaluation. The only experts available were associated with lab where AuraOrb was developed. While this may have lead to bias in the results, to perform the evaluation, it was necessary that the experts be recruited. As well, it may be beneficial in the future to include a sample of participants who are not very familiar with electronic devices and computers. While the number of participants was sufficient to uncover a number of usability problems, it may be possible to uncover more problems had more usability experts been recruited.

In the future it may be of benefit to compare newer computer-based notification techniques to AuraOrb. For example, Microsoft has updated their email client to provide a peripheral dialog box which displays the sender and subject. As well, AuraOrb's ambient visual notification could be compared to ambient auditory notifications to determine which output modality is found to be less disruptive.

Handling of multiple notifications could be improved by sorting notifications based on user-defined priorities. Alternatively, multiple orbs could be deployed for different types of notifications. For example, one orb could be configured to notify the user of incoming emails, while another orb could alert the user of stock price changes. A third orb could inform the user if their items on eBay are outbid. In this manner, the user can be made aware of changes from multiple sources without the need for manual interactions.

As well, the touch sensing in AuraOrb could be improved to allow the detection of gestures. This could allow richer interactions and may be another way to improve the access to multiple queued notifications. The user could move their hand across the surface of the orb to quickly scroll through notifications. As well, gestures could be assigned to quickly delete emails or file them in pre-assigned email boxes, directly from AuraOrb.

6.4 Conclusion

In this thesis, we presented AuraOrb, an ambient notification display that deploys progressive turn taking techniques. AuraOrb uses an eye contact sensor to detect user interest in an initially ambient light display. When eye contact is detected, AuraOrb displays the subject heading of the notification. Touching the orb causes the associated message to be displayed on the user's last attended computer screen. When user interest is lost, AuraOrb automatically reverts back to its idle state. The design of AuraOrb was guided by our observations on previous work and three design principles that we identified as requirements for the development of ambient and progressive notification appliances.

We conducted a heuristic evaluation that compared AuraOrb's progressive notification techniques to a scrolling ticker tape display and a GUI system tray email notification. Almost all evaluators preferred using AuraOrb to handle their notifications. They commented that the fluid transition to information content upon eye contact allowed them to more quickly evaluate whether to accept or ignore the notification. This supports our belief that the use of progressive turn taking techniques leads to a more sociable and less disruptive notification strategy.

Bibliography

[1] P. Adamczyk and B. Bailey. If not now, when?: The effects of interruption at different moments within task execution. In *Proceedings of CHI 2004*, pages 271–278. ACM Press, 2004.

[2] V. Bellotti, M. Back, W.K. Edwards, R. Grinter, A. Henderson, and C. Lopes. Making sense of sensing systems: Five questions for designers and researchers. In *Proceedings of CHI 2002*, pages 415–422. ACM Press, 2002.

[3] V. Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, July 1945.

[4] A. Chang, B. Resner, B. Koerner, X. Wang, and H. Ishii. Lumitouch: An emotional communication device. In *Extended Abstracts of CHI 2001*, pages 313–314. ACM Press, 2001.

[5] S. Consolvo and J. Towle. Evaluating an ambient display for the home. In *Extended Abstracts of CHI 2005*, pages 1304–1307. ACM Press, 2005.

[6] Ambient Devices. Ambient orb. `http://www.ambientdevices.com`, 2005.

[7] A. Dix, J. Finlay, G. Abowd, and R. Beale. *Human-Computer Interaction.* Prentice-Hall Europe, second edition, 1998.

[8] J. Fogarty, S. Hudson, C. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. Lee, and J. Yang. Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction*, 12(1):119–146, 2005.

[9] W. Gibbs. Considerate computing. *Scientific American*, January 2005.

[10] E. Horvitz, A. Jacobs, and D. Hovel. Attention-sensitive alerting. In *Proceedings of UAI 1999*, pages 305–313. Morgan Kaufmann Stockholm, 1999.

[11] E. Horvitz, C. Kadie, T. Paek, and D. Hovel. Models of attention in computing and communication. *Communications of the ACM*, 46(3), 2003.

[12] G. Hsieh, K. Wood, and A. Sellen. Peripheral display of digital handwritten notes. In *Proceedings of CHI 2006*. ACM Press, 2006.

[13] Phidgets Inc. Unique usb interfaces. `http://www.phidgets.com`, 2005.

[14] Wave — Olympia Industries. Infoglobe. `http://www.soundbug-us.com`, 2005.

[15] S. Iqbal and B. Bailey. Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption. In *Extended Abstracts of CHI 2005*, pages 1489–1492. ACM Press, 2005.

[16] H. Ishii, C. Wisneski, S. Brave, A. Dahley, M. Gorbet, B. Ullmer, and P. Yarin. ambientroom: Integrating ambient media with architectural space. In *Extended Abstracts of CHI 1998*, pages 173–174. ACM Press, 1998.

[17] J.C.R. Licklider. The computer as a communication device. *Science and Technology*, April 1968.

[18] P.P. Maglio and C.S. Campbell. Tradeoffs in displaying peripheral information. In *Proceedings of CHI 2000*, pages 241–248. ACM Press, 2000.

[19] A. Mamuji, R. Vertegaal, J. Shell, T. Pham, and C. Sohn. Auralamp: Contextual speech recognition in an eye contact sensing light appliance. In *Extended Abstracts of Ubicomp 2003.* ACM Press, 2003.

[20] J. Mankoff, A. Dey, G. Hsieh, J. Kientz, S. Lederer, and M. Ames. Heuristic evaluation of ambient displays. In *Proceedings of CHI 2003*, pages 169–176. ACM Press, 2003.

[21] T. Matthews, A. Dey, J. Mankoff, S. Carter, and T. Rattenbury. A toolkit for managing user attention in peripheral displays. In *Proceedings of UIST 2004*, pages 247–256. ACM Press, 2004.

[22] D.S. McCrickard and C.M. Chewar. Attuning notification design to user goals and attention costs. *Communications of the ACM*, 46(3), 2003.

[23] D.S. McCrickard, C.M. Chewar, J.P. Somervell, and A. Ndiwalana. A model for notification systems evaluation – assessing user goals for multitasking activity. *ACM Transactions on Computer-Human Interaction*, pages 312–338, 2003.

[24] D.S. McCrickard, M. Czerwinksi, and L. Bartram. Introduction: Design and evaluation of notification user interfaces. *International Journal of Human-Computer Studies*, 8(5), 2003.

[25] Inc. MicroEngineering Labs. Caller id. `http://www.melabs.com`, 2005.

[26] T. Miller and J. Stasko. Infocanvas: Information conveyance through personalized, expressive art. In *Extended Abstracts of CHI 2001*, pages 305–306. ACM Press, 2001.

[27] MouseSite. Doug engelbart 1968 demo. `http://sloan.stanford.edu/mousesite/1968Demo.html`.

[28] E. Mynatt, J. Rowan, S. Craighill, and A. Jacobs. Digital family portraits: Supporting peace of mind for extended family members. In *Proceedings of CHI 2001*, pages 333–340. ACM Press, 2001.

[29] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *Proceedings of CHI 1990*, pages 249–256. ACM Press, 1990.

[30] A. Oh, H. Fox, M. Van Kleek, A. Adler, K. Gajos, L. Morency, and T. Darrell. Evaluating look-to-talk: A gaze-aware interface in a collaborative environment. In *Extended Abstracts of CHI 2002*, pages 650–651. ACM Press, 2002.

[31] J. Shell, T. Selker, and R. Vertegaal. Interacting with groups of computers. *Communications of the ACM*, 46(3), 2003.

[32] J. Shell, R. Vertegaal, and A. Skaburskis. Eyepliances: Attention-seeking devices that respond to visual attention. In *Extended Abstracts of CHI 2003*, pages 770–771. ACM Press, 2003.

[33] Whirlwind Software. Cidmage. `http://www.codegods.net`, 2005.

[34] M. van Dantzich, D. Robbins, E. Horovitz, and M. Czerwinski. Scope: Providing awareness of multiple notifications at a glance. In *Proceedings of AVI 2002*. ACM Press, 2002.

[35] R. Vertegaal. Designing attentive interfaces. In *Proceedings of ACM ETRA Symposium on Eye Tracking Research and Applications 2002*. ACM Press, 2002.

[36] R. Vertegaal. Attentive user interfaces. *Communications of the ACM*, 46(3), 2003.

[37] R. Vertegaal, R. Slagter, G. van der Veer, and A. Nijholt. Eye gaze patterns in conversations. In *Proceedings of CHI 2001*, pages 301–308. ACM Press, 2001.

[38] D. Vogel and R. Balakrishnan. Interactive public ambient displays: Transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proceedings of UIST 2004*. ACM Press, 2004.

[39] R. Want, K. Fishkin, A. Gujar, and B. Harrison. Bridging physical and virtual worlds with electronic tags. In *Proceedings of CHI 1999*, pages 370–377. ACM Press, 1999.

[40] M. Weiser. The computer for the 21st century. *Scientific American*, September 1991.

[41] M. Weiser. The future of ubiquitous computing on campus. *Communications of the ACM*, 41(1), 1998.

[42] Wikipedia. History of computing. `http://en.wikipedia.org/wiki/History_of_computing`.