# EyeReason and EyePliances: Tools For Driving Interactions Using Attention-Based Reasoning

by

Aadil Aziz Mamuji

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 2005

*In the Name of God, the Most Gracious, the Most Merciful*

~·~

*Oh Allah, increase me in knowledge, but let this knowledge be with sincerity - not seeking glory, status, or material wealth. Let this knowledge serve Your cause in a way that You accept, and let it benefit humanity.*

# Abstract

The prevailing ubiquity of computers has brought about a considerable increase in the number of digital appliances at our disposal. Such ubiquitous appliances, however, are still designed to act in isolation. Each appliance may independently notify the user of incoming communications or computer activity, without any consideration to the user's engagement or interaction with other devices or persons. Devices continue to relay volumes of email, instant messages, phone calls and appointment notifications, collectively producing an intricate web of annoying 'attention grabbers', within which a user can easily become quite entangled.

This research proposes that by coordinating communications on the basis of user activity, availability, or more specifically user attention, devices may engage in more polite and respectful interaction with their users, without fragmenting their limited cognitive attention [48, 63]. Although designers of computer systems may be able to develop notification strategies that are less disruptive and better coordinated between devices and users, interruptions generated by computer systems are not the only source of interference with the user's focus task. Active human ad-hoc and co-located group communications may be equally distracting and problematic. This research explores how

regulating communications by ubiquitous sensing and reasoning about the focal point of the user's communications may, in the future, alleviate such problems in homes and office scenarios.

This is explored through the design of Attentive User Interfaces. Firstly, a personal communication server called EyeReason, which acts as a central receptionist that handles all direct and indirect interactions of a user with computers, and secondly, through the design of computing appliances, called EyePliances, which are sensitive to a user's attention. Our methods are analogous to human-turn taking in group communication. Just as turn-taking improves an individual's ability to conduct foreground processing of conversations [65], Attentive User Interfaces bridge the gap between the foreground and periphery of user activity, thereby facilitating users to move smoothly in between. By sensing a user's attention for objects and people in their everyday environment, and by treating user attention as a limited resource, EyeReason and EyePliances work together to avoid today's ubiquitous patterns of interruptions. Instead, they take turns for communication, by sensing when the user is paying attention to them before taking the floor.

# Collaborators

The design, development and idea of EyeReason and all EyeReason applications as a possible solution to attention-based management of interactions are my contributions towards this research. However, there are a number of individuals who have generously applied their expertise within all of its constructs. Their efforts will not go unrecognised or unappreciated.

My supervisor, *Roel Vertegaal,* formulated the Attentive User Interface paradigm, and together with *Jeffery Shell* conceptualized the foundations of EyeReason and EyePliances [34, 50, 63]. *Changuk Sohn* applied master engineering skills to design and produce the eye contact sensors. In addition, he developed the hardware in all the prototype applications [33-35, 62]. *Daniel Cheng* improved and implemented the image processing algorithms for eye contact detection [35, 49, 62]. *Thanh Pham* was instrumental in the initial development and testing of EyeReason and the application prototypes [34]. *Maria Danninger* is accredited with developing the Social Geometry toolkit [10] used by the Attentive Cubicle [33]. *Connor Dickie* used his background in film to produce and edit the videos made for conference submission [33].

# Acknowledgements

All praise be to Allah (God), the most Glorious, the most High. To Him, I am eternally thankful, for everything.

To my parents, sister and other family, I owe everything else. Their endless love and prayers have provided me with the comfort and strength to keep going.

A sincere thank you to my supervisor for his guidance; mentorship; and snappy dressing. And, an animated "word" goes to all my colleagues at the Human Media Lab, for the good times; for the support; and for shining that much needed light to lead me through – mad respect.

A final word goes to my fiancé. For sharing in the smooth times and the rough; for carrying me forward; and for that special place in your heart, I love you.

# Contents

# List of Tables

# List of Figures

# Chapter 1

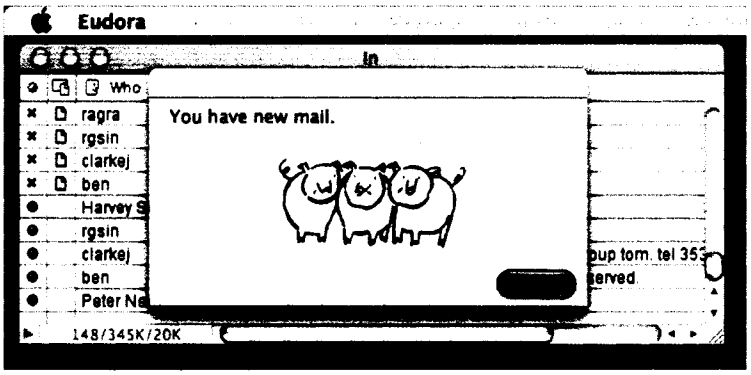
# Introduction

## 1.1 Human Computer Interaction (HCI)

The proliferation of ubiquitous digital devices necessitates a new way of thinking about Human-Computer Interaction (HCI). Weiser [67] said of Ubiquitous Computing: "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." Although our society lives in a ubiquitous computing age with technologies that have seamlessly interwoven themselves into our daily existence, the interface has effectively not disappeared. For many years HCI design and research efforts have focused on the development of computers as tools that are essentially extensions of analog devices such as paper, pencils and typewriters. While this view will continue to prevail for years to come, our

community is now beginning to see limits to this approach. One reason for this is that unlike traditional tools, computers are becoming increasingly *active* communicators. They are, however, ill equipped to articulate their communications towards their users.

The example in Figure 1-1 below serves as a clear illustration of this dilemma. Without *any* regard for the user's current activity, a modal dialog box pops up in the *center* of the screen, alerting its user that a message has been received. Only by clicking the *"OK"* button can the user continue her activities. This example highlights the serious underlying flaw in user interfaces: the computer's inherent lack of knowledge about the present activities of its user. The behaviour of such devices may indeed be described as being socially inadequate.




The modal e-mail notification alert

**Figure 1-1.**
"You have new mail"

Meier showed early on that interruptions that distract a user from a focus task are an important source of work-related stress [39]. Experiments conducted by Einstein et al. further indicated that demanding work conditions as well as frequent interruptions revealed rapid forgetfulness of intentions at levels that would be considered significant in applied settings. They also noted that presenting a user with a visual notification, such a

a small blue dot in the lower right-hand corner of the screen, enabled participants to completely overcome the negative effects of interruptions [14].

Hudson et al. [28] conducted a Wizard of Oz study to explore how robust sensor-based predictions of "interruptibility" might be constructed, and how useful they might be to such predictions. They concluded that sensor-based estimators of human interruptibility are possible and that they can achieve results within a 75-80% accuracy range. They found that a relatively simple set of sensors could be employed to achieve good results [28].

## 1.2 HCI: A Multiparty Dialogue

As new relationships with computing systems evolve, integrate into and surround daily life, the above studies show that there is a need to develop better strategies for design. Society has moved on from many users sharing a single computer through a command line interface, to many persons using many computers that are all interconnected. In essence, a *multiparty* relationship has developed between users and their computers, causing existing channels of interaction to break down because:

- Each user is surrounded by *many* active computing devices.
- These devices form part of a worldwide inter-connected network.
- Users form part of a worldwide "attention seeking" community through these active devices.

Given the prevalence of actively connected devices, users are now being bombarded with interruptions from their Palm Pilots, BlackBerries, Smart Communicators, email programs, auction trackers, instant messaging tools and cell phones. Like the pop-up

email notification in Figure 1-1 the nature of interruptions is usually acute, irritating and invariably requiring immediate attention. Consequently, user attention has become a limited resource, continually vied for by various devices, each claiming high priority. We believe that instead, computers should be designed with channels that explicitly negotiate the volume and timing of communications with their user, pending the user's current needs. Our design strategy to solving this problem, by making interfaces more considerate [17] and less interruptive, rests upon the most striking parallel available: that of multiparty dialogue in *human group communication.*

## 1.3 Human Group Communication

In human group conversation, attention is inherently a limited resource, as humans can only listen to and absorb the message of one person at a time [9] [13]. Interestingly, humans have exhibited two ways of coping with interference from multiple conversational sources of information. By using nonverbal cues that convey attention, humans achieve a remarkably efficient process of speaker exchange, or *turn-taking* [13]. Turn-taking provides a powerful metaphor for the regulation of communication with ubiquitous devices. According to Short et al. [52], as many as eight cues may be used to indicate an upcoming exchange of turns:

i. completion of a grammatical clause;
ii. a socio-centric expression such as 'you know';
iii. a drawl on the final syllable;
iv. a shift in pitch at the end of the clause;
v. a drop in loudness;

vi. termination of gestures; and

vii. relaxation of body positions

viii. the resumption of eye contact with a listener.

In group conversations, however, only eye contact indicates to *whom* the speaker may be yielding the floor [59]. Eye contact indicates with about 82% percent accuracy whether a person is being spoken or listened to in four-person conversations [64]. When a speaker falls silent and looks at a listener, this is perceived as an invitation to take the floor. According to a recent study, 49% of the reason why someone speaks may be explained by the degree of eye contact made with an interlocutor [61]. Humans use eye contact in the turn taking process for four reasons:

i. Eye fixations provide the most reliable indication of the target of a person's attention, including their conversational attention [2] [64].

ii. The perception of eye contact increases arousal, which aids in proper allocation of brain resources, and in regulating inter-personal relationships [2].

iii. Eye contact is a *nonverbal visual* signal, one that can be used to negotiate turns without interrupting the *verbal auditory* channel.

iv. Eye contact allows them to observe the nonverbal responses, including the attentional focus, of others.

Conversational turn-taking, however, is typically deployed in formal contexts such as during meetings [13]. Here the speaking behavior of others can be controlled through social protocol. By asking only one speaker to be active at any one time, the act of turn-taking allows each listener in the meeting to focus the limited attentional resources of their brain on to a single speaker. However, there are situations in which conversational

turn-taking is either undesirable or impractical. In public transportation, coffee shop or cubicle farm scenarios, conversational activity of others cannot be controlled and may possibly present interference to others. In these situations the human brain copes by attenuating irrelevant auditory stimuli, a process known as the Cocktail Party phenomenon [9]. Here the brain uses both environmental and semantic conversational stimuli to tune its attentive system to a single cohesive message from a single conversational source.

Proximity and social orientation of users is also useful for determining when users *are* interested in communicating with one another. However, determining when a user is engaged in conversation with another person is not a straightforward problem. According to Edward Hall's theory of proximity in communicative space [23] interaction with other persons takes place within a certain distance range that varies with culture. Proximity encompasses intimate, personal, social and public space. Intimate space is used for touching one another, personal space supports conversations, social distances involve groups of people, and public spaces take in the wider context.

## 1.4 Measuring Attention

According to Maglio et al., the above behaviours transfer to scenarios where users interact with devices [32], in that users tend to look at a device when issuing a command [32]. Based on the above exploration of turn taking behavior in human group conversations, we believe eye gaze sensing may provide a viable means of detecting user interest for communications with a device. There are a number of reasons why the use of

eye gaze as a means for conveying attention is compelling, over other methods such as pointing:

i. In mobile scenarios, users do not need to carry an input device to perform basic pointing tasks. In scenarios where the hands are busy or otherwise unavailable, eye gaze provides an extra and independent channel of input.

ii. The eyes have the fastest muscles in the human body, and consequently are capable of moving much quicker than any other body part. Also, researchers have reported that during target acquisition, users tend to look at it *before* initiating manual action [31]. This means that if tracked effectively, eye gaze could provide one of the fastest possible input methods.

iii. Users can produce thousands of eye movements without any apparent fatigue. Eye gaze mitigates the need for repetitive manual actions, and thus reduces the risk of repetitive strain injury.

iv. Users are *very* familiar with the use of their eyes as a means for selecting the target of their commands. They use eye gaze during their communications with other humans to effectively indicate whom they are addressing or listening to. Users are also familiar with others responding to them whenever they make eye contact [64].

It is, however, important to distinguish between the use of the eyes as a continuous pointing device and for selecting discrete targets. Users are *not* very familiar with the use of their eyes as a continuous pointing device, essentially because eyes provide input to the human body, rather than output to control the exterior environment. Furthermore, eyes do not typically perform well in continuous pointing, because in order to inhibit the

impact of the world's movement on the retina, they naturally move very rapidly between fixation points [12].

## 1.5 A Midas Touch

There are also other arguments against eye tracking as an input device.

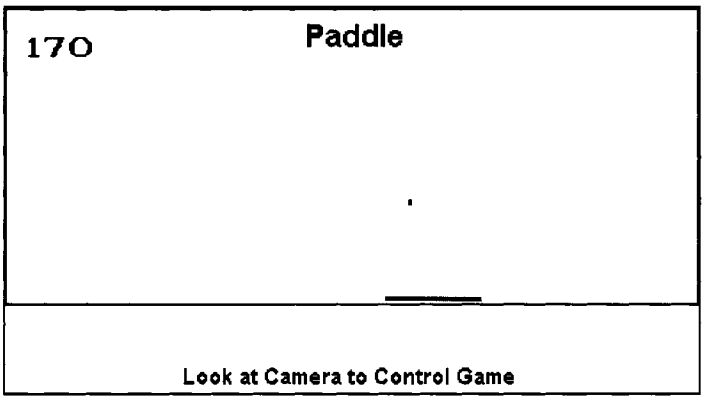
i. Intolerance to head movement. The history of eye tracking has produced some gruelling contraptions that essentially kept the user's head motionless. Advances in computer vision, however, now enable users to move relatively freely, with commercial trackers capable of achieving head movement tolerances of over 30x15x20 cm.

ii. Eye tracking can be inaccurate and noisy. On-screen accuracies of better than 1 degree are now the norm, and further improvements are likely. These however are still considerably less effective than that of manual pointing techniques,. It has been suggested by Jacob [31] and others that the accuracy of eye trackers in pointing is fundamentally limited by the size of the human fovea, which is in the order of 2° of visual angle [12]. This argument suggests that there would be no need for the eye to position with greater accuracy than what is required to keep a visual target within the fovea. It may be noted though that inaccuracies are actually caused by current limitations in existing computer vision algorithms [12].

iii. Eye trackers are expensive, mainly because of low market demand.

iv. Until recently, eye trackers needed to be calibrated by having users track predefined targets.

v. Eye trackers suffer from what is known as the Midas Touch Effect [31]. The Midas Touch Effect is caused by overloading the eye's visual input function with a motor output task and occurs chiefly when an eye tracker is used not only for pointing, but also for clicking. Clicking with the eyes is useful when users do not have control over limbs. In such cases, the Midas Touch effect causes users to inadvertently select or activate any target they fixate upon. By issuing a click only when the user has fixated on a target for a certain amount of time (*dwell time click*), the Midas Touch effect can be controlled, though not entirely removed. The effect, however, can be avoided by issuing clicks via an alternate input modality, such as a manual button or voice command. More generally, if the output task interferes with the input task, the effectiveness of eye tracking input is greatly reduced. When mapping input to the eyes, it is therefore important to select cases whereby the movement of the eyes for the output task matches that required for visual input. A great example of such a scenario is the use of an eye tracker to play Pong (see Figure 1-2).




The paddle tracks the horizontal coordinate of the eye. Here, as users are observing the ball, the horizontal coordinate of their eye movements is used to automatically move the paddle. The chief exploit of Pong, the eye-hand control problem is thus mitigated, making eye-controlled Pong a game one cannot lose!

**Figure 1-2.**
Eye-controlled Pong.

## 1.6 Problem Statement and Contributions

In this thesis, we explore the development and use of eye contact sensing as a means for providing information about the user's focus of attention to smart appliances. The eye gaze of the user, as an extra channel of input, in many cases is an ideal candidate for ubiquitous devices to sense when their user is paying attention to them, to another device or to a person.

This thesis also discusses the potential of the paradigm of Attentive User Interfaces. Attentive User Interfaces are systems that sense, analyse and optimize the user's attention and mitigate interruptions directed towards the user. There are two components to this system. First is the development of a personal communication server called EyeReason, which acts as a central receptionist that handles all *remote* interactions of a user, including computers and people [50, 63]. Second is the development of computing or home appliances that are sensitive to a user's attention, called EyePliances. EyePliances are attentive gaze and speech enabled household devices, such as lamps and home theatre systems that are augmented with a miniature eye tracker called an Eye Contact Sensor (ECS) [51]. The ECS reports to the EyeReason server whenever that user is engaged with an associated appliance. Together, these Attentive User Interfaces work in methods similar to human-turn taking in group communication. Just as turn-taking improves an individual's ability to conduct foreground processing of conversations [65], Attentive User Interfaces bridge the gap between the foreground and periphery of user activity, thereby facilitating smooth movement between the two. Focusing upon attention as a central interaction channel allows development of more sociable methods of communication with devices around us.

By sensing a user's attention for objects and people in their everyday environment, and by treating user attention as a limited resource, EyeReason and EyePliances work together to avoid today's ubiquitous patterns of interruptions. More significantly, they take turns for communicating by sensing before taking the floor whether the user is paying attention to them.

## 1.7 Overview

This thesis begins at Chapter 2 with a literature review of the Attentive User Interface paradigm, followed by a presentation, in Chapter 3, of the contribution that this thesis offers to the field of HCI. In Chapter 4 and 5 it continues with an outline of a number of prototype applications that have been developed to support this research, primarily focussing on three broad areas of interaction: human-device; human-human; and human-group interactions. Finally, it concludes in Chapter 6 with a discussion on initial user experiences as well as thoughts for future consideration in Chapter 7.

# Chapter 2

# Literature Review

## 2.1 The Attentive User Interface Paradigm

The management of interruptions generated by communication technologies has recently become an important topic of study in HCI [27] [28]. Bellotti et al. [6] posed five challenges for multiparty HCI and this thesis hopes to provide some suggestions towards answering at least three of these:

i. How do I address one of many possible devices?

ii. How do I know the system is ready and attending to my actions?

iii. How do I specify a target for my actions?

How *do* people move from GUI-style interactions, where multiple entities are represented on a single computing device, to interactions with many remote devices? For one, it is important to note that many of the elements of GUIs were designed with attention in mind. According to Smith et al. [53], *windows* provide a way to optimally allocate the available screen real estate to accommodate user task priorities. Windows represent foreground tasks at high resolution, and occupy the bulk of display space in the center of vision. *Icons* represent peripheral tasks at low resolution at the edges of the user's vision. *Pointers* allow users to communicate their focus of attention to graphic objects. By clicking on icons to open windows, and by positioning, resizing and closing windows, users use their pointing device to manually manage their attention space. By control-clicking graphic objects, users indicate the target of menu commands. In clicking *"OK"* buttons, users acknowledge interruptions by alert boxes (see Figure 1-1 on page 2).

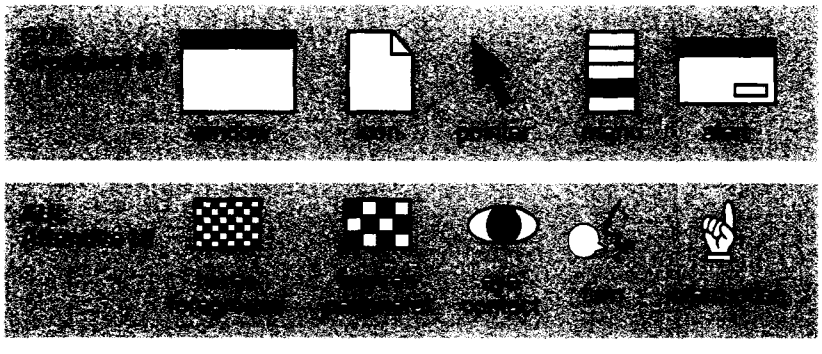


**Figure 2-1.**
Equivalents of Graphical UI elements in Attentive UI.

Figure 2-1 shows how we might extend these GUI elements to interactions with ubiquitous remote devices, drawing parallels with the role of attention in human turn-taking. Windows and icons are supplanted by graceful increases and decreases of information resolution between devices in the foreground and background of a user's attention landscape. Devices sense whether they are in the focus of user attention by

observing presence and eye contact; and menus and alerts are replaced by a negotiated turn-taking process between the users and devices. Such characteristics and behaviours define an Attentive User Interface (AUI).

AUIs aim to recognize a user's attention space in order to optimize the effectiveness of the information processing resources and devices within his or her physical domain. This is accomplished by measuring and modeling the users' past, present and future attention for tasks, devices or people. Five key features of AUIs include [48] [63]:

i. *Sensing attention:* By tracking users' physical proximity, body orientation and eye fixations, interfaces can determine what device, person or task a user is most likely attending to.

ii. *Reasoning about attention:* By statistically modeling simple interactive behavior of users, interfaces can estimate the user's task prioritization.

iii. *Communication of attention:* Interfaces should make available information about the users' attention to other people and devices. Communication systems should convey to whom or to what users are paying attention, and whether a user is available for communication.

iv. *Gradual negotiation of turns:* Like turn taking, interfaces should determine the availability of the user for interruption by a) checking the priority of their request; b) progressively signaling this request via a peripheral channel; and c) sensing user acknowledgment of the request before taking the foreground.

v. *Augmentation of focus.* The ultimate goal of all AUIs is to augment the attention of their users. Analogous to the Cocktail Party Phenomenon, AUIs may, for

example, magnify information focused upon by the user and attenuate peripheral detail.

This work was inspired by interactions with a host of researchers, designers, media artists, the vision of Ubiquitous Computing [67], as well as by the seamless interaction created by considering foreground versus background in Tangible User Interfaces [29]. Since these paradigms are well known, we will limit the discussion to existing Attentive User Interfaces and describe examples that relate to our framework. We present work in two areas of research: attention management in single user interactions with computing devices and in co-located social group interactions.

## 2.2 Attention Management in Single User Interactions

We begin our overview with a look at the one-on-one relationship between a user and his or her computer.

### 2.2.1 Sensing Attention: *Eye Tracking as a Tool*

Rick Bolt's *Gaze-Orchestrated Dynamic Windows* was one of the first true AUIs [7]. It simulated a composite of 40 television episodes playing simultaneously on one large display. All stereo soundtracks from the episodes were active, creating "a kind of Cocktail Party Effect mélange of voices and sounds". Via a pair of eye tracking glasses, the system sensed the user's visual attention towards a particular image, turning off the soundtracks of all other episodes; and zooming in to fill the screen with the focal image. Bolt's system demonstrated how a windowing system could be translated into a display with malleable resolution that exploits the dynamics of the user's visual attention. It

shows the great potential of AUIs to augment attention by reducing information overload in congested audio-visual environments. Jacob [30] and MAGIC [69] showed that eye tracking works best when it is applied to observe user attention, rather than as a device for control. This is because, to the user, the eyes are principally an input rather than an output organ. As a consequence, when the duration of an eye fixation on an on-screen object is used to issue commands, users may unintentionally trigger unwanted responses while looking (The *Midas Touch* effect [30]). In a *Non-Command Interface* [44] version, instead of a user explicitly issuing commands, the computer observes user activity. The system then *reasons about action* using a set of heuristics. In the classic game of Paddleball, the goal is to position a sliding paddle into the path of a moving ball using a joystick, which in turn introduces an eye/hand coordination problem. In a Non-Command Interface version of the game, the paddle location is given by the horizontal coordinate of a user's on-screen gaze, communicating the visual attention of the user and thus eliminating the game's eye/hand coordination problem (see Figure 1-2 on page 9).

### 2.2.2 Reasoning about Attention

Horvitz et al.'s Attentional Interfaces [26] use Bayesian reasoning to identify what channels to use and whether or not to notify a user. In the *Priorities* system [26], the delivery of email messages is prioritized using simple measures of user attention to a sender: the mean time and frequency with which the user responds to emails from that sender. Messages with a high priority rating are forwarded to a user's pager, while messages with low priority wait until the user checks them. Attentional Interfaces are characterized by their ability to *reason* about user attention as a resource, rather than *sense* attention for a device.

### 2.2.3 Negotiating Turns

Simple User Interest Tracker (*SUITOR*) [32] was one of the first Attentive Information Systems. *SUITOR* provides a GUI architecture that tracks the attention of users through multiple channels, such as eye tracking, web browsing and application use. It uses this to model the possible interest of the user, presenting suggestions and web links pertaining to the task at hand. In order not to interfere with the user's foreground task, it displays all suggestions using a small ticker tape display at the bottom of the screen. SUITOR shows the importance of modeling *multiple* channels of user behavior; and demonstrates how to use a peripheral low-density display to avoid interrupting a user with information, the relevance of which to the foreground task is not fully known.

*Pong* is a robot head that rotates to face users by tracking pupils with a camera located in its nose [41]. *FRED* [64] is an Attentive Embodied Conversational System that uses multiple animated head models to represent agents on a screen. Agents track eye contact with a user to determine when to take turns. Pong and FRED show how anthropomorphic cues from head and eye activity may be used to signal device attention to a user, and how speech engines can track eye contact to distinguish what entity a user is talking to. FRED shows how proximity cues may be used to move from foreground to peripheral display with malleable resolution. When the user stops talking and fixating at an agent, the agent looks away, and shrinks to a corner of the screen. When users produce prolonged fixations at an agent and start talking, the agent makes eye contact and moves to the foreground of the display.

Maglio et al. [32] and Oh et al. [45] demonstrated that when issuing spoken commands, users do look at the individual devices that execute the associated tasks. This

means eye contact sensing can be used to open and close communication channels between users and remote devices, a principle known as *Look-to-Talk.* EyeR [47] is a pair of tracking glasses designed for this purpose. By emitting and sensing infrared beams, these glasses detect when people orient their head towards another device or user with EyeR. EyeR, however, does not sense eye position, and this stimulated Shell et al. to develop eye trackers suitable for Look-to-Talk: low-cost, calibration-free, long range and wearable *eye contact sensors* (see Section 3.1) [49].

### 2.2.4 Augmenting Attention: *Less is More*

Attentive focus through multi-resolution vision is a fundamental property of the human eye. The acuity of the human retina is highest at the *fovea*, a 2° region around the visual axis. Beyond 5°, visual acuity drops into *peripheral vision* [12]. *Gaze-contingent Displays* update their images in between fixations to allow alignment of visual material with the position of the fovea, as reported by an eye tracker. Originally invented to study vision, reading and eye disease, gaze-contingent displays now help to optimize graphics displays [38] [12]. By matching the level-of-detail of a 3D graphic card rendering with the resolution of the user's eye, Virtual Reality display have improved [43].

With the move towards Context-Aware Interfaces [40], we are seeing increased use of attentive visualization in HCI. *Focus+Context* [5] is a wall-sized low-resolution display, with a high-resolution embedded display region. Users move graphic objects to the high-resolution area for closer inspection, without losing the context provided by peripheral vision. It is an elegant example of *static* multi-resolution windows. *Popout Prism* [56] focuses user attention on search keywords found in a document by presenting keywords throughout a document in enlarged, colored boxes. Such Attentive User

Interfaces are distinct from Context-Aware Interfaces in that they *focus* on designing for attention.

Architects and designers such as Mies Van Der Rohe [8], have long advocated focusing design resources in ways that provide synergies between manufacturing, human factors and aesthetic requirements. His adagio "Less is More" reflects the need to consider human attention in design. Many tools can be characterized as having been designed with attentive properties in mind. The thin blue lines that aid handwriting on paper are a good example. Since peripheral vision is least sensitive to blue detail, the lines are visible only when you need them [12].

According to Goldhaber [19], the Internet can be viewed as an economy of attention. Drawing analogies with human group communication, Goldhaber convincingly argues that buying and selling attention is its natural business model. Indeed, advertising agencies sell page views, while the *Google* [20] search engine ranks results by the number of outside links to a page.

## 2.3 Attention Management in Co-located Group Interactions

With the availability of ubiquitous computers and public wireless access points comes an increase in the use of computing systems in public environments. Whether it is on public transportation, in coffee shops or cubicle farms, users increasingly work in environments in which the level of environmental distraction cannot be controlled. Focusing on the task at hand can be highly problematic when working in a public space. There have been a number of interfaces designed to support the user's attentional strategies for coping with environmental sources of distraction.

A prevalent source of noise is presented by the conversational activity of others within the shared space. In busy office and public environments, active ad-hoc and co-located group communications among co-workers may distract others from completing their tasks. In general, while the Cocktail Party phenomenon [9] helps us filter extraneous noise, it is by no means a perfect process. According to Gillie and Broadbent, this attentive mechanism is especially sensitive to disruption by information that is semantically related to the ongoing task [18].

There are, to date, few examples of systems that measure participant attention in order to manage interruption during co-located meetings. Stiefelhagen [54] developed a system that tracked head orientation of participants using computer vision and a neural network during four-person meetings. Although computationally expensive, Stiefelhagen's system has been successfully applied to automated editing of meeting recordings [55].

Auditory stimuli present a particular challenge because of their omni-directional nature. McFarlane suggested giving users control over the delivery of auditory stimuli, as, according to him, this considerably improved task performance [37]. In this regard, Basu and Pentland presented a pair of Smart Headphones that detected and relayed sounds in the environment through to the user's headset, but only if they were classified as human speech [4]. Mueller and Karau's Transparent Headphones helped users listen to digital music while still being accessible to surrounding individuals [42].

Erickson and Kellogg introduced the notion of "social translucence". They argued that socially translucent technologies were fundamental requirements for supporting all types of human-human communication [15]. To allow for social translucence it is critical that information about the orientation of body, head and eyes of co-located individuals be sensed [28, 64]. By mounting proximity sensors on the Transparent Headphones, the system detected when a person approached a user, presumably to engage in conversation. However, eye-gaze is a much more accurate predictor of conversational engagement between individuals. In a busy subway station, for example, there may be many people walking in close proximity to the wearer. In such situations, Transparent Headphones would be likely to frequently block input signals [42].

While Mueller and Karau experimented with the use of infrared transceivers, they did not sense eye gaze. More importantly, their headphones did not offer TiVo®-like features such as buffering and fast-forwarding of real-world conversations [58]. Reference may be made to Deitz and Yerazunis who discuss their use of real-time audio buffering techniques to manage periods of distraction in telephone conversations [11]. While the phone handset is away from the user's ear, incoming audio is recorded in a circular

buffer. Two pointers are used to indicate where to start and stop accelerated audio playback. Using time-compression and pitch preservation algorithms, they allow users to quickly catch up to real-time phone conversations without the loss of information [11].

The framework presented in this thesis extends upon the basic principles outlined by these HCI researchers, to create attention aware systems that truly augment the user's attention – thereby augmenting his or her mental power to better manage incoming information from multiple, often simultaneous sources.

# Chapter 3

# Sensing Attention

A number of techniques have been developed to aid in the sensing of where a user is looking and his or her active participation in human social groups.

## 3.1 Sensing Eye Contact

An eye contact sensor (ECS) is essentially an inexpensive eye tracker that detects whether a person is looking at the sensor or not. It requires no prior calibration of any kind. Shell et al. designed a sensor that can be built cheaply, consisting of a camera that finds pupils within its field of view using a simple computer vision algorithm [49]. The ECS consists of an infrared camera with a set of on-axis infrared LEDs mounted around the camera lens (see Figure 3-1). When flashed, these produce a *bright pupil* reflection (similar to the "*red eye*" effect caused by photographic camera flashes) in eyes within

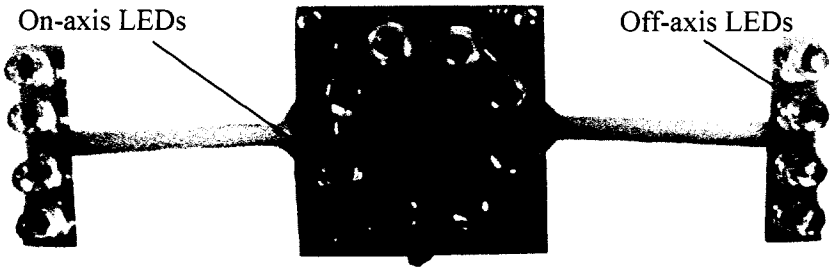



**Figure 3-1.**
Eye Contact Sensor

range. Another set of LEDs is mounted off-axis away from the camera lens. Flashing these produces a similar image, with black pupils. By synchronizing the LEDs with the camera clock, a bright and dark pupil effect is produced in alternate fields of each video frame. A simple algorithm finds any eyes in front of the camera by subtracting the even and odd fields of each frame [49]. The LEDs also produce a glint on the cornea of the onlooker's eyes. These appear near the center of the detected pupils when the onlooker is looking straight at the camera, allowing the detection of eye contact. When mounted on a device, the eye contact sensor obtains information about the number and location of pupils in its field of view, and whether these pupils are looking at the sensor. Via a high level communication protocol, it reports this information wirelessly over a TCP/IP connection to a connected EyeReason server. ECS data is typically filtered by EyeReason, with eye contact reported only when the amount of gaze over time exceeds a user-defined threshold.

Eye contact sensors are cheap eye tracking input devices especially designed for the purpose of implementing Look-To-Talk with ubiquitous appliances. Unlike traditional eye trackers, their only requirement is to detect the user looking straight at the device.

## 3.2 Social Proximity and Identification

In addition to sensing eye contact towards a wearer, ECSs also allow the sensing of social proximity information. According to Hall there are four zones of social proximity: *intimate* (0.0 - 0.45 m); *personal* (0.45 - 1.2 m); *social consultative* (1.2 -3.0 m); and *public* (over 3.0 m). Most conversational activity occurs within a personal zone of social proximity [22]. ECSs can determine social proximity cues by measuring the distance between detected sets of pupils.

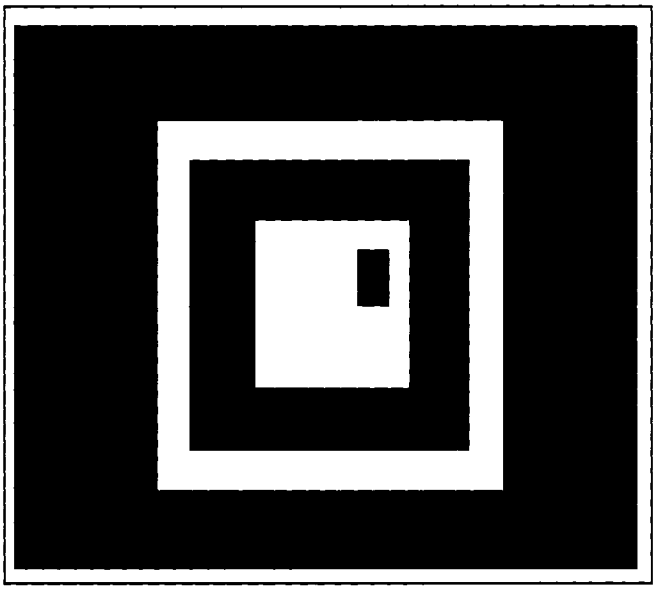
The ECS calculates the approximate distance of an onlooker by determining his Interpupillary Distance, and comparing this measure to a known mean of 6.2 cm in a general population [22]. Eye contact sensors can also be used to uniquely identify other eye contact sensors [49]. This is accomplished by using one of the infrared LEDs on the ECS to send a unique binary identifier through a pulse code modulated infrared beam. This is used to allow an attentive headphone to detect *who* is looking at their wearer.

## 3.3 Tracking Head Orientation in Large Groups

While eye contact between individuals is one of the most direct and reliable measures of engagement between two individuals [64], head tracking provides a more tractable problem when dealing with large groups [55]. We therefore adopted the Social Geometry toolkit developed by Maria Danniger [10], that uses head orientation to determine joint attention between individuals across wider areas, through simple overhead computer vision. The use of an overhead camera provides the additional advantage of making head location data readily available. Studies show that head

orientation provides a reasonable estimate of a person's direction of regard, accurate to within 15 degrees of visual angle [54].

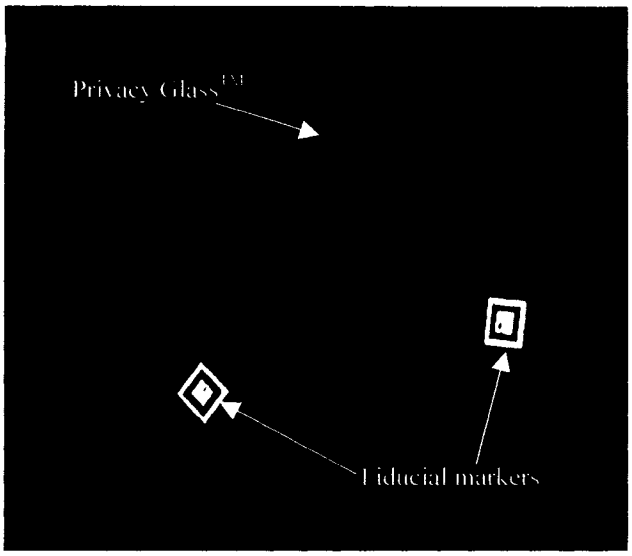
To allow for scalability, real-time performance and reliability at low cost, the social geometry tracking system is based on the ARToolKit [3]. The ARToolKit is a software library for augmented reality applications that can be used to calculate camera position and orientation relative to fiducial[1] markers in real time. The ARToolKit was adapted to allow tracking of multiple moving targets from a stationary camera. The system used fiducial markers that are distinctively recognized from one another; that are tolerant to tilt; and that have an asymmetric pattern (see Figure 3-2).



Fiducial Markers used for tracking head location and orientation

**Figure 3-2.**
Fiducial Marker

[1] Fiducial markers are selected patterns in an image that are used as a frame of reference in locating objects. They are uniquely identifiable.



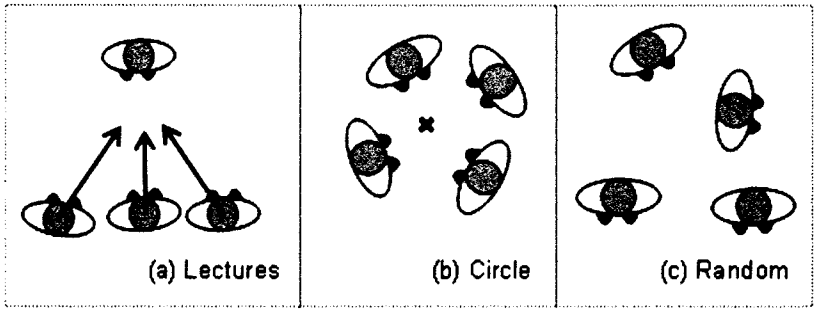

View from overhead camera. Retroreflective Fiducial markers on people's headsets ease the motion capture process.

**Figure 3-3.**
Retroreflective Fiducial markers on people's headsets.

To ease the motion capture process, a webcam augmented with infra-red LED illuminators was used. Figure 3-3 shows the image from a camera located in the ceiling of an office environment. The users wear fiducial markers, affixed with retroreflective strips (that are easily detected under infra-red light) attached to headsets. This facilitates identification of head location and orientation in two dimensions and at very low camera resolution.

To establish social group membership from individual head movement measurements, the Social Geometry system performs analysis of the "social geometries" formed by virtual connections between a group of co-oriented heads. By the established definition, individuals may share mutual attention if their heads are oriented towards each other for a certain minimum duration and located within a certain maximum social distance from one another [2]. Rather than determining group geometries on a frame-by-frame basis, the system uses a dynamic approach that relies on statistical definitions of co-orientation over time [10]. One of the defining features of social groups is that they

typically present geometric clusters of people interacting with each other. This is graphically illustrated in Figure 3-4 below.




Examples of 4 persons in different group geometries.

**Figure 3-4.**
Examples of Social Groups

The examples show how the structure of these clusters may allow classification into group types based on relatively simple geometric properties of relationships between individual bodies. The illustrations identify the one-to-many or lecture geometry (Figure 3-4a), where most members are orientated towards a single speaker; and the many-to-many arrangement of round table meetings (Figure 3-4b). In both these geometries, the orientation and proximity of participants leads to a clustering that is quite distinct from the arbitrary grouping shown in Figure 3-4c [10].

By mounting eye contact sensors on multiple ubiquitous devices, eye fixations can be tracked with great accuracy throughout the user's environment. Furthermore, the social geometry clusters illustrated in Figure 3-4 of individual users provide a powerful yet computationally inexpensive method of group analysis. This effectively allows EyeReason the use of fine grained measures such as eye gaze and coarse grained measures such as body orientation to reason about a user's attention for devices and membership within a social group.

# Chapter 4

# EyeReason And EyePliances

## 4.1 Reasoning About Attention

Eye Contact Sensors (ECS) allow any home appliance to sense when users are looking at them, without requiring any form of calibration. This allows users to determine, simply by looking at the appliance, which appliance is currently the target of remote control commands. We call such gaze and speech enabled household appliances EyePliances [49] [51].

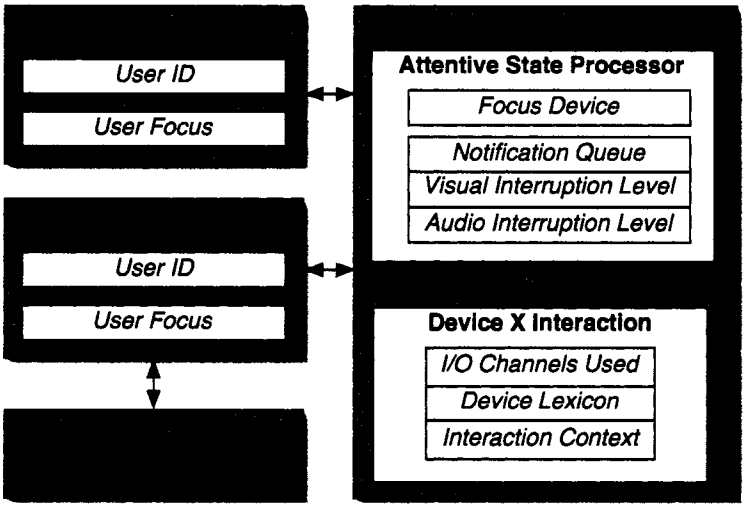
Each EyePliance is connected to a central server known as EyeReason, which switches commands between EyePliances upon user eye contact. When an EyePliance is used in conjunction with other EyePliances, commands that can be issued by voice, remote control or Bluetooth keyboard, can be easily reused amongst devices. When a

remote control is used, its commands are interpreted by the server and relayed to the appliance that the user looks at through an RF [24], X10 [68], or infrared transmitter interface.

The chief advantage of this approach is that it allows users to control a number of appliances without having to select from many buttons, and without placing the remote control in a device-specific modality. In the case of voice recognition, the user need not carry an input device at all, as basic commands can be issued to a speech recognition engine located on the EyeReason server. Upon eye contact with an EyePliance, this speech recognition engine switches its lexicon to that of the focus EyePliance. After a command is interpreted, it is relayed to the appliance.

The EyeReason system coordinates communications among many EyePliances and the user by keeping track of user activity with each device. It operates as a centralized server that EyePliance clients may connect to. By tracking manual interactions and eye contact, devices report to the server whether a user is actually working with them. When the EyeReason system determines a device is in the focus of user attention, it raises the priority of communications between that device and the user, and typically allows the device with the highest priority to take the floor. When a speech recognition EyePliance takes the floor, EyeReason turns on its speech engine and switches the lexicon to that of the focus device. Requests from competing EyePliances may be suppressed by EyeReason, or routed to a device within the user's focus depending on the content of that information [34] [50] [63]. Figure 4-1 below shows the EyeReason architecture for each user.

## 4.2 EyeReason Architecture




Each user has an associated EyeReason server that co-ordinates interactions between him or her and connected devices

**Figure 4-1.**
EyeReason Architecture.

EyeReason employs a plug-and-play capability that maintains a list of connected devices. Each device is assigned a corresponding Device Driver Client – a virtual representation of the device – that relays commands and notification messages between the device and EyeReason. Upon connection, the Device Driver Client is initialised with a unique number that allows EyeReason to identify the device. Each identification number is assigned randomly within a specific range, depending on what category the device falls under (see Table 4-1 below). A check is implemented to ensure that each ID is unique to each instance of a device, and is only available again after the device disconnects from the server. If a device wishes to connect with the same ID as assigned previously, it can do so as well. Once assigned an ID, the Device Driver Client informs EyeReason of the message notification capabilities of the corresponding device (see Figure 4-2 on page 34).

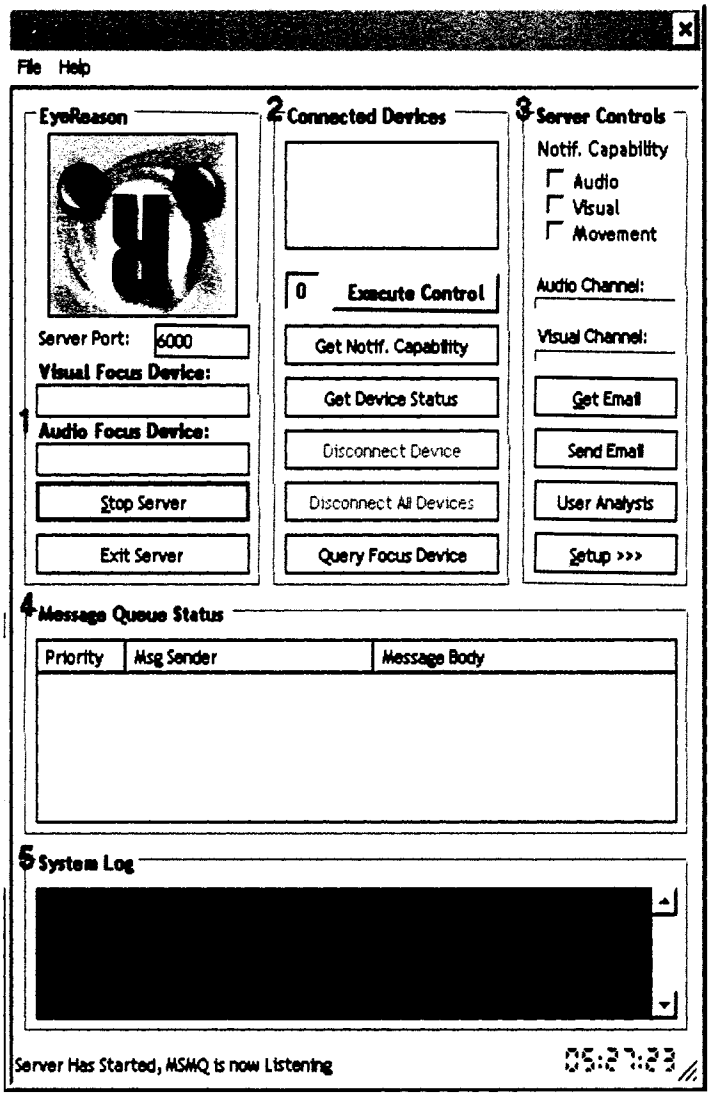
| Device | ID range |
|---|---|
| smartCubicle | {1000,...,1999} |
| smartEmail | {2000,...,2999} |
| smartLight | {3000,...,3999} |
| smartHeadphones | {4000,...,4999} |
| smartTelevision | {5000,...,5999} |
| smartTelephone | {6000,...,6999} |

Each new instance of a device is assigned a random value within the specified range.

**Table 4-1.**
Device Identification Range.

When a user interacts with a particular device for a prolonged period of time, the EyeReason server determines that it is a device within the user's focus, and is known as the focus device. Requests from competing devices to deliver information may either be suppressed by the server, or routed to the focus device depending on the content of that information. In the case of incoming email, the server can determine the priority of the message using a Bayesian model, similar to that employed by Horvitz in the Priorities System [26]. In the case of speech interaction, devices need to be in the focus of user attention before the system allows the user and device to converse. By opening and closing communication channels on the basis of user-device interaction, EyeReason acts as a gatekeeper determining which device should be allowed to take the floor.

EyeReason, provides a facility to coordinate communications among EyePliances by modeling the user's visual or auditory attention for devices. Figure 4-2 below is a screenshot of the EyeReason server.




1. Displays the current device that the user is looking at or listening to.

2. Displays the devices that are currently connected to EyeReason. Commands can be sent to these devices to query device status, get notification capabilities, execute control commands remotely, as well as to convey messages to the user.

3. Displays the notification capabilities of the selected device as well as the user's level of engagement with the audio and visual channel of the device (if applicable). Based on User Analysis, this allows EyeReason to determine the less distracting method of relaying messages and notifications to the user.

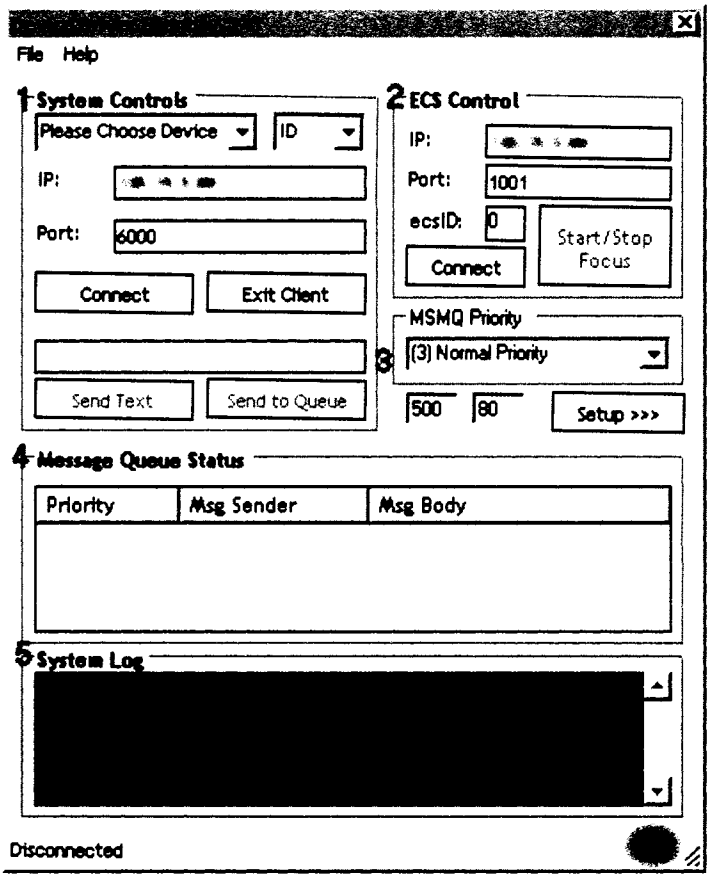
4. Messages received from remote devices are placed in a queue here. Depending on which device a user is interacting with, a message's priority determines how and when it is relayed to the user.

5. Keeps a real-time log of all activities between EyeReason and connected devices.

**Figure 4-2.**
Screenshot of the EyeReason Server.

## 4.3 Gaze Activated Controls

The EyeReason architecture simplifies the process of augmenting a standard appliance with gaze and speech capability. By embedding an eye contact sensor in an appliance and specifying an appropriate XML speech grammar, a device instantly becomes an EyePliance. Upon connection to EyeReason, each EyePliance is assigned a "virtual" Device Driver Client that relays commands between EyeReason and the appliance. Figure 4-3 shows a screenshot of a typical Device Driver Client.




1. Before connection, a Device Driver Client is assigned to an EyePliance. An ID (for identification within EyeReason) can be assigned by default or an existing ID can be used.

2. The Device Driver Client connects to the ECS associated with the EyePliance and reports eye contact.

3. Messages sent to EyeReason are assigned a priority.

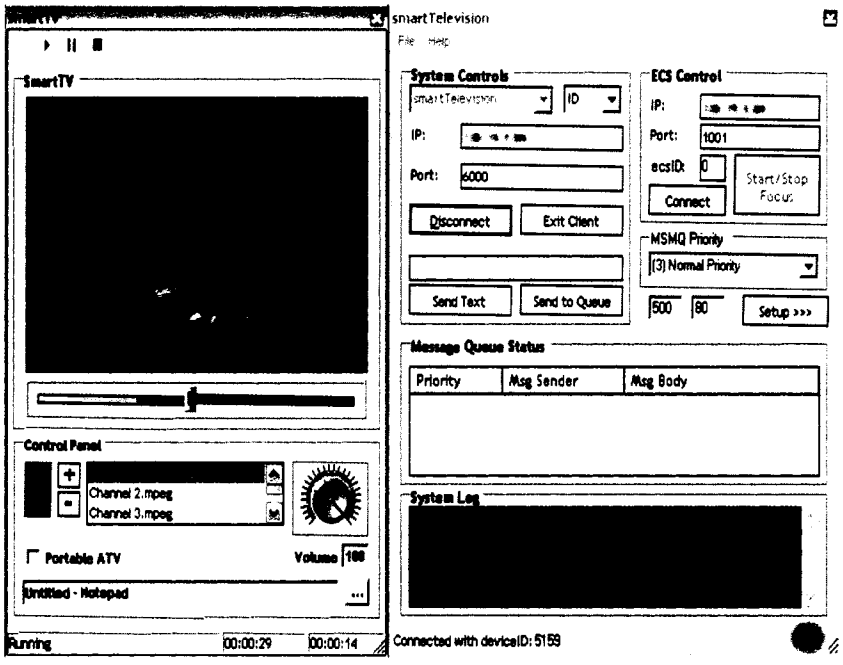
4. Messages received from EyeReason are stored in an MSMQ (see section 4.1.3).

5. Keeps a real-time log of all activities between the client and EyeReason.

**Figure 4-3.**
Screenshot of a Typical EyePliance Driver.

If the EyePliance receives eye contact, the Device Driver Client informs EyeReason that it is now the device with visual focus and processes speech commands from a wireless headset using the XML lexicon specified in EyeReason. The tasks that are subsequently performed can either be activated through remote control, X10, Bluetooth keyboard, or through direct interfacing into the appliance. If none of these are available, EyeReason still recognizes that a user is engaged with the device. Figure 4-4 shows a screenshot of a Device Driver Client assigned to interface with a Television EyePliance.

## 4.4 EyePliances and EyeReason Interactivity




A Device Driver Client is a "virtual" representation of an EyePliance. It bridges communications between an EyePliance and EyeReason.

**Figure 4-4.**
Screenshot of a smartTelevision.

Communication between EyePliances and EyeReason is accomplished via TCP/IP. Through the corresponding virtual Device Driver Client, specific communication protocols (see Appendix A) between the client and server allow EyePliances to inform

EyeReason of the current activity a user is engaged in. For example, if a user is watching the Television, its virtual Device Driver Client (in this case called smartTelevision) which is within EyeReason's architecture, determines that the user is engaged in a visual as well as an auditory task with the TV. If the user mutes audio from the TV to answer a phone call, but continues to maintain eye contact, the smartTelevision reports that the user is only occupied with a visual task. The Telephone also has a Device Driver Client associated with it (called smartTelephone) which reports that the user is engaged in an audio task with it (see Figure 4-5 below). Each channel of interaction depends on the task associated with the EyePliance. The level of the audio channel reflects the volume of the device, while the visual channel reflects whether the user is looking at the device or not. This constitutes the attentive status of the user.

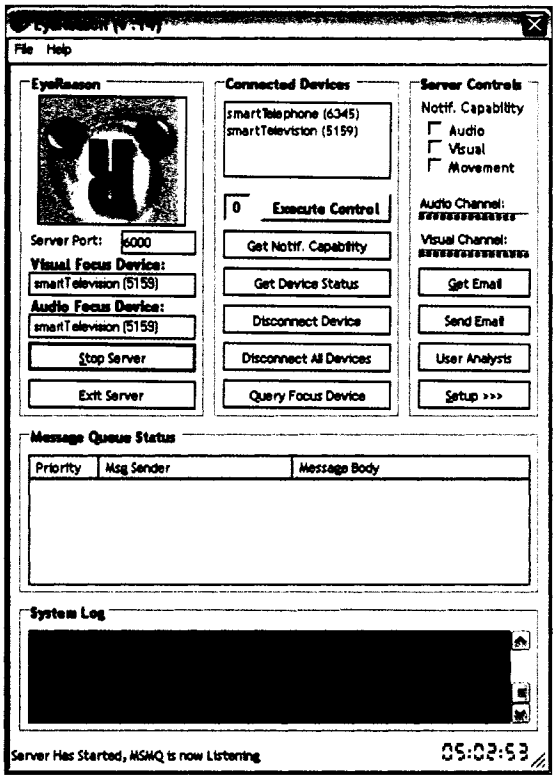



Figure 4-5a. - Indicates that both the visual and auditory channels of the television are occupied.

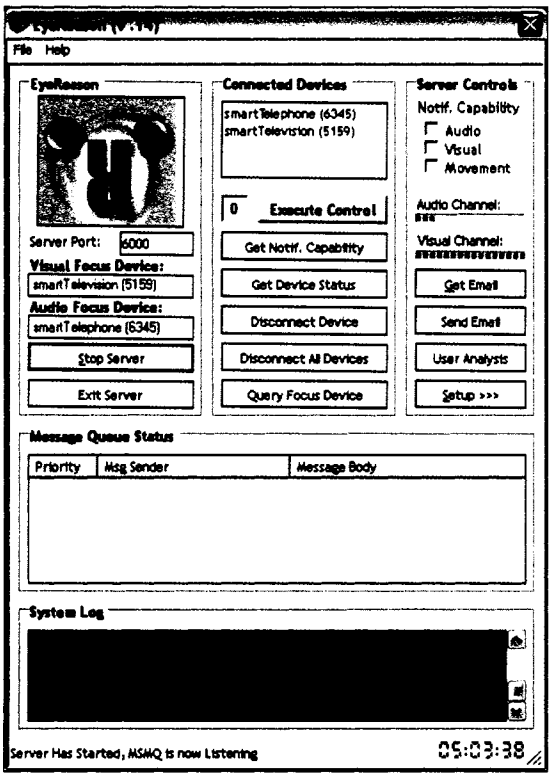


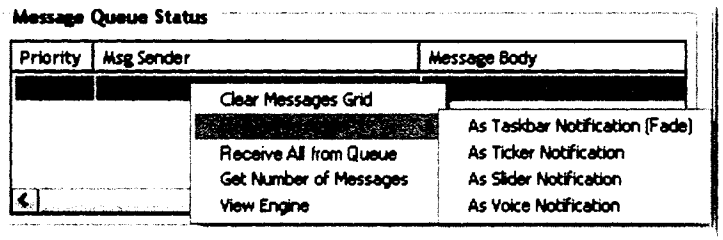

Figure 4-5b. - Indicates that the *visual* channel of the television is occupied while the *auditory* channel of the telephone is occupied.

**Figure 4-5.**
Screenshots of EyeReason receiving information on user's interaction with EyePliances.

EyeReason can relay messages between EyePliances; execute control commands to an eyePliance; or query an EyePliance about its current state. It is equipped with a Microsoft Messaging Queue (MSMQ) Service which is a tool for sending and receiving messages. Each Device Driver Client has a queue, where messages can be retrieved in a prioritized first-in first-out manner. Each Device Driver Client assigns a priority to a message before sending it to EyeReason. For each message, EyeReason determines the location and method of delivery. Figure 4-6 below shows the delivery options for messages.

### 4.4.1 Message Notifications

Numerous empirical evaluations have been conducted to determine the most effective method of notification. McCrickard et al. showed that if the goal was to identify items quickly (i.e. of high priority), an in-place display like a fade or blast should be used, while if the goal was to increase comprehension, a motion-based display like a ticker should be used [36].




The MSMQ service in EyeReason appears with the available delivery options for a message. EyeReason determines where a message should be delivered based on what EyePliance a user is currently engaged with. The method of delivery is based on the priority of the message as well as whether the visual or audio channel of the device is occupied.

**Figure 4-6.**
Microsoft Messaging Queue (MSMQ)

We incorporated these findings into how EyeReason delivers message notifications to the user. Based on the priority of a message, and the current EyePliance a user is attending to, message notifications are delivered in either of four ways:

i. A blast and fade notification

ii. A ticker notification

iii. A slider notification

iv. A voice notification

The message deliveries are designed to present as little interruption to the user's current task as possible. The blast and fade, ticker and slider notifications are used when the EyePliance reports eye contact, and are typically deployed when the audio channel of an EyePliance is occupied. This helps to minimise interruption to the user's cognitive auditory load. The voice notification is usually used when the user is not looking at any particular EyePliance, or when he is engaged with an EyePliance that is not capable of displaying messages. Voice notifications are rarely used since they prove to be the highly interruptive to a user's current task [18]. They are typically deployed on very high priority messages that require immediate attention.

While a user is engaged with an EyePliance to perform a particular task, EyeReason can also communicate the user's attentive status to his or her list of personal contacts through their Attentive Cell Phone [60]. Depending on the relationship between the user's buddy and the set of current tasks, EyeReason can determine the importance of a contact. For example, in a work scenario, an employee's manager would have elevated

interruptive priority over social contacts. In the case of emergency situations these levels can be explicitly overridden.

## 4.5 EyeReason: Attention-Based Management

By mounting eye contact sensors on multiple ubiquitous devices, eye fixations can be tracked with great accuracy throughout the user's environment. Also, the social geometry clusters illustrated earlier in Figure 3-4 (on page 29) of individual users provide a powerful yet computationally inexpensive method of analysis. It effectively allows the use of coarse grained measures such as body orientation to reason about attention for and membership of a social group.

The EyeReason server is ideally located on a user's portable device, such as a PDA or Blackberry. EyeReason wirelessly networks all eye contact sensors in a room to monitor attention towards EyePliances, and reasons about social group membership based on straightforward geometrical and temporal properties of mutually shared attention between people. This allows a user's interaction with devices as well as membership in social groups to be monitored and regulated, based on his or her attention capacity. The EyeReason system coordinates communications among many devices and the user by keeping track of the interaction with each device. In addition, for each tracked person within a social group, the Social Geometry engine reports information about potential communication partners to that individual's EyeReason server [50] [34], as given by the ID of the fiducial marker. By opening and closing communication channels on the basis of human-device interaction, and human-human interaction, EyeReason serves as a digital receptionist that actively manages a person's attention.

# Chapter 5

# EyeReason Applications

EyePliances are smart ubiquitous appliances with embedded attention sensors, designed to extend the existing concept of gradual turn taking. Users interact with EyePliances through speech, keyboard, radio tags [66] or manual interaction. Functionality in appliances is accessed through the X10 home automation software [68] and wireless Internet connectivity. Section 5.1.2 discusses the simplest form of an EyePliance – a light fixture appliance with an embedded eye contact sensor [34]. As a goal, Attentive User Interfaces emphasize the design of interactions such that they optimize the use of the user's attentive resources. We will now describe our efforts towards the development of a number of Attentive User Interface prototypes.

## 5.1 Using EyeReason To Manage Human-Device Communications: EyePliances

We discuss two EyePliance prototypes here: AuraLamp and EyeTuner. AuraLamp is a lava lamp augmented with an eye contact sensor and speech recognition capabilities [34]. As the accuracy of speech recognition interfaces increases, we believe users will come to rely more on voice commands in their interactions with such appliances. However, without specific naming conventions, speech recognition engines cannot determine which device, among many, the user is speaking to.

AuraLamp listens to simple voice commands such as "On" or "Off", but only when the user focuses his attention on the lamp. EyeTuner essentially consists of a speaker with a digital ECS mounted on top allowing the speaker to sense when users are looking at it [62]. Both examples, demonstrate how we may coordinate communications between a user and many ubiquitous appliances by sensing when the user pays attention to a particular device. Rather than competing for the user's attention, devices enter a turn taking process similar to that used in human group conversation.

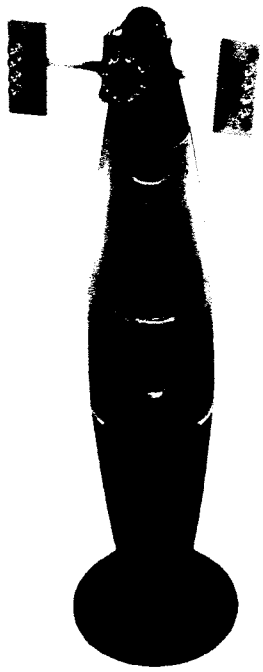
In this section, we discuss the design of ubiquitous appliances that use the eye gaze of the user to determine when to communicate. By augmenting ubiquitous devices with eye contact sensors that determine when the user looks at them, appliances obtain knowledge about the current engagement of a user with the device. Such information not only aids in the use of deictic references in speech or remote control interfaces, it also provides a significant source of information for determining when devices should *avoid* communications with their user [48] [63].

### 5.1.1 Gaze Activated Speech Lexicons

As speech commands are processed through the centralized EyeReason server, new forms of attentive interactivity are permitted without increasing the complexity of each appliance. With the Look-to-Talk paradigm as a foundation, EyeReason acts as more than just a gatekeeper for interactions with ubiquitous appliances. It integrates a speech recognition system that dynamically activates the control context of the device as the user shifts focus. The gaze actuated speech recognition encapsulated in EyeReason eliminates contextual ambiguity when interacting with a device via a voice channel. Since EyeReason allocates voice control only to the EyePliances currently in focus, it allows duplicate voice grammar definitions across devices.

EyeReason uses the Microsoft Speech API 5.1 SDK to implement these context-sensitive grammars through XML-based lexicons. Processing speech to AuraLamp, for example, through EyeReason involves two steps. First, the AuraLamp device driver detects activity information representing the attention of the user by polling the associated eye contact sensor over a TCP/IP connection. When a sufficient level of eye contact is detected, the driver loads the EyePliance's context specific grammar. When an EyePliance driver activates its grammar, EyeReason automatically deactivates grammars for EyePliances not in the focus of user attention.

### 5.1.2 AuraLamp



AuraLamp light fixture with embedded eye contact sensor.

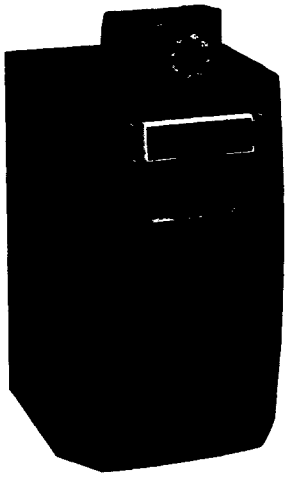
**Figure 5-1.**
AuraLamp Light Fixture

The AuraLamp (Figure 5-1) is a lava lamp augmented with an eye contact sensor and speech recognition capability. By looking at the lamp, a person indicates attention to the device, thereby activating its speech engine. When the user does not look, its speech engine deactivates and does not listen to the user. This avoids problems of multiple appliances listening at the same time, removing ambiguity in user speech command interpretation. Since only one appliance is the active listener, users can use deictic references when referring to the device. Having only one of several appliances to be the active listener allows the use of a single centralized speech recognition engine. This in turn greatly reduces the speech processing load for the total set of appliances.

AuraLamp responds only to the two actions it is capable of: turning on and turning off. By switching the active speech recognition lexicon on the EyeReason server to that of the EyePliance currently in focus, the accuracy of speech recognition is increased, while at the same time presenting the user with a small reusable vocabulary. AuraLamp is a model for how we may use visual attention with speech to interact with any household appliance.

Each speech command in the lexicon is associated with an X10 home automation command. A serial interface routes these commands from the speech processing server to the electricity grid [68]. Over standard electrical wiring, the commands reach a simple controller unit capable of turning the appliance on or off. The X10 interface makes it easy to extend our interaction model to any appliance in the household.

### 5.1.3 EyeTuner



AirTunes Speaker with Digital ECS.

**Figure 5-2.**
EyeTuner

The EyeTuner shown in Figure 5-2 essentially consists of a speaker with digital ECS, mounted atop, that allows the speaker to sense when users are looking at it. This speaker is connected over an AirTunes network to a computer running Apple's iTunes [1]. Whenever users produce a prolonged fixation at the speaker, our central EyeReason server responds by lowering the volume of the song currently playing. If eye contact is sustained, it starts parsing user commands, whether issued by remote control, Bluetooth Keyboard, or voice commands through a lapel microphone.

Apart from recognizing standard remote control commands such as play, pause and skip, users can also query the iTunes library for tracks. Queries are performed using the Bluetooth keyboard, or through speech recognition. Users issue a speech query by saying "Find <name>" while looking at the speaker. Upon receiving the "Find" command, the speech engine switches its lexicon to the names of individual tracks, albums and artists within the user's iTunes library. The <name> query is subsequently submitted to iTunes

over a TCP/IP connection. If a query results in multiple hits, EyeTuner responds by showing the list on its LCD display [46], after which it begins playing the first track. Users can subsequently skip through the tracks until the desired song is located.

## 5.2 Using EyeReason To Manage Human-Human Communications: Attentive Headphones

To experiment with the use of EyeReason as a means of managing human colocated communications we developed the Attentive Headphones: a pair of noise cancelling headphones augmented with a microphone and an ECS [35].

### 5.2.1 Attentive Headphones

The main problem of today's noise-cancelling headphone is that it creates an attentional barrier between users. This barrier reduces social translucence [15] to the wearer of the headset, as auditory signals for attention by co-workers come to be ignored. When users wearing noise-cancelling headsets in cubicle farms were observed, it was noticed that these devices essentially offer an all-or-nothing strategy for coping with environmental noise. Users either have their headset engaged and are working on a computer task, or they are in a conversation with their headphones off. More importantly, it was found that co-workers frequently have problems approaching a headphone user with sociable requests for attention. Because headsets filter out all environmental stimuli, when users are focused on their computer screen, they may not even notice the presence of a co-worker. As a consequence, co-workers often resort to shoulder taps and other physical means of requesting attention. The problem with this is that it typically crosses the boundaries of social rules of engagement [15].

Here, we discuss the design of a noise-cancelling headset that is sensitive to social requests for attention. We augmented a pair of headphones with eye contact sensors that detect when someone looks at their wearer, both from behind and from the front. The headphones are also equipped with a microphone that picks up the wearer's voice. Upon detecting eye gaze directed at the wearer from behind, the headsets automatically turn off noise-cancellation. This provides an ambient notification that allows users to decide whether to attend to any request for attention using normal social protocol.

## 5.2.2 Implementation

Figure 5-3a – *Right view*

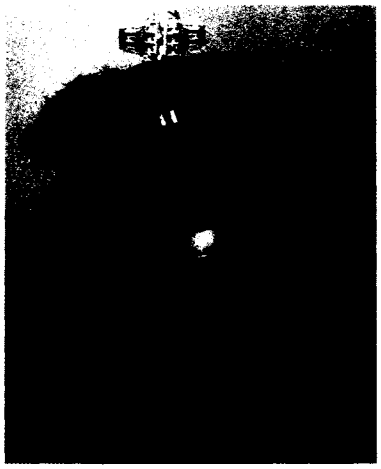


Figure 5-3b – *Left view*

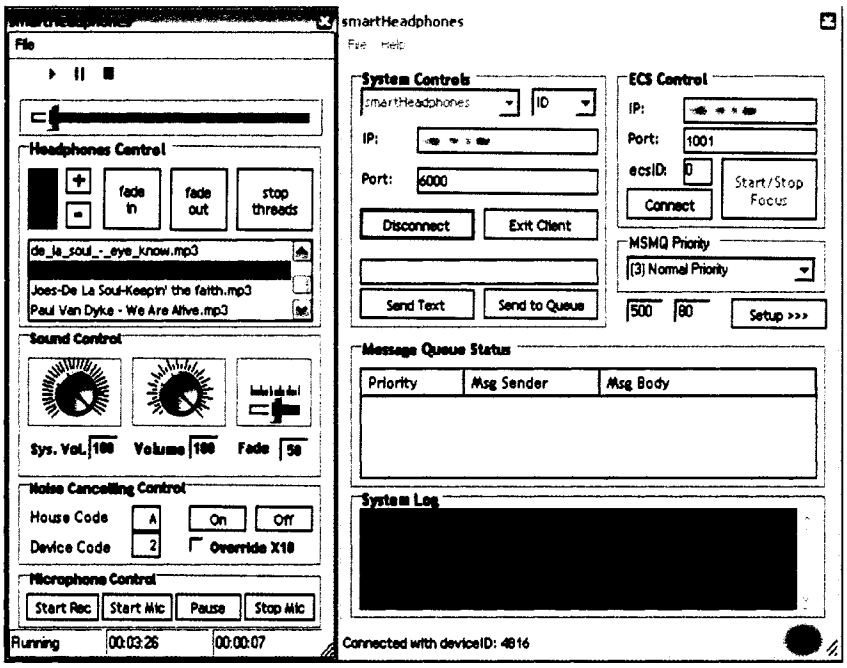
Attentive Headphones with embedded eye contact sensors on the front and back, and lapel microphone

**Figure 5-3.**
Attentive Headphones

An Attentive Headphone consists of a Bose™ noise-cancelling headphone augmented with two eye contact sensors, pointing to the front, and the back respectively, as well as a lapel microphone (see Figure 5-3). We modified the headset with a circuit that allows noise-cancellation to be switched on or off wirelessly through an X10 interface [68]. When the headset is turned off, wearers can hear each other normally. When the headset

is turned on, ambient sound is attenuated by –20 dB[2]. Sound from the microphone is sent through a wireless connection to a server that buffers the sound and relays it to other headsets. When the wearer is engaged in a computer task, visual requests for attention outside the wearer's field of view are detected by an eye contact sensor on the back of the headset (Figure 5-3b). Griffin and Bock showed that participants tended to fixate on a given entity in a scene roughly 900 milliseconds before verbally referring to it [21]. To avoid unintentional triggering, the back ECS only reports fixations that are longer than 1 second. This time interval can be adjusted manually according to user preference. Similarly, the headset can detect when the wearer is in a conversation by polling the second ECS, mounted toward the front of the headset This ECS scans the eyes of individuals standing in front of the wearer in order to predict when the wearer is likely to be engaged in conversation [49]. The front ECS reports only on pupils within about 1-2 meters, the *personal* social proximity zone [22]. The information from multiple ECSs is integrated through the user's personal EyeReason server [49]. The EyeReason server determines which device or person the user is likely to be engaged with, by polling all eye contact sensors associated with that user. Figure 5-4 below illustrates the Device Driver Client for the Attentive Headphones, called smartHeadphones.

[2] dB is the logarithmic units used to describe sound intensity (or amplitude). dB is used to indicate the loudness of a sound. The larger the number, the louder the sound.




The Device Driver Client for the Attentive Headphones.

**Figure 5-4.**
Screenshot of smartHeadphones

## 5.2.3 Attentive Headphones Operation

When the headphones detect eye contact by an onlooker, the EyeReason server responds by temporarily turning off noise cancellation on the headset, pausing any audio currently playing in the headset This allows the voice of the potential interlocutor to be heard by the wearer, and the request to be serviced according to social protocol. It also functions as a subtle ambient notification of the pending request for attention. The user's EyeReason server determines when the user responds to the request by detecting eye contact with people in front of the user within a user-defined interval. When eye contact is not detected within that period, noise-cancellation is again engaged, and any audio playing on the headset is resumed. When eye contact *is* detected, noise cancellation is left off instead, allowing the wearer to have a normal conversation with the interlocutor. If the user ends the conversation and returns to his task, this is detected by loss of eye

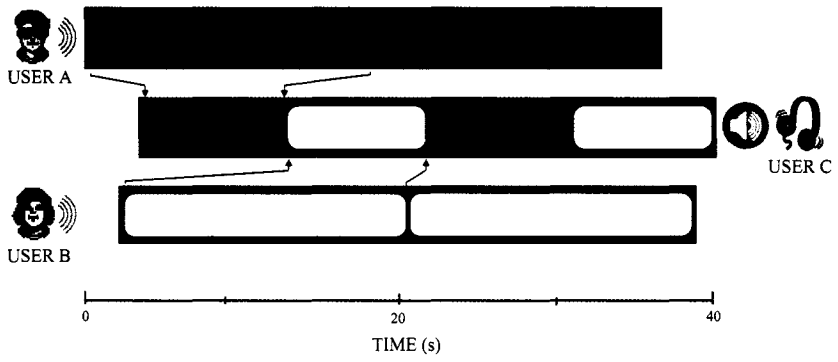
contact with the frontal ECS. When this occurs, headset noise cancellation is engaged and any content previously playing in the headphones smoothly fades in, and continues where it was paused.

Even without noise cancellation, the headphones tend to attenuate sound from the outside world. To alleviate this problem, sound from the microphone mounted on the headset can be relayed to other headsets within the current social network, as determined by eye contact between headset wearers. While this further improves the signal to noise ratio of sound from attended individuals, we did not want to limit the operation of the headphones to enhanced cocktail party filtering.

We were particularly interested in experimenting with ways in which we could boost the user's attentional capacity. To achieve this, we experimented with the use of buffering techniques similar to those of a TiVo® personal video recorder [58]. For this purpose, each wearer's EyeReason server continuously records audio streams from individuals engaged within his or her social network. A button on the headset allows users to pause live conversational audio, for example upon receiving a cell-phone call. This allows them to attend to the call without losing track of the ongoing conversation. Pressing the button a second time plays back the recorded conversation at double speed, without affecting its apparent pitch [11]. Buffering can be set to trigger automatically upon servicing an incoming phone call.

### 5.2.4 Attending to Two Simultaneous Speakers

We also experimented with the use of time multiplexing techniques that would allow users to attend to two speakers at once. When two individuals, A and B, are within a user C's current social network and begin talking simultaneously, user C's EyeReason server begins an automated turn-taking process in which it plays back recorded speech from A and B at twice the speed in a time-multiplexed fashion. Since the voices from user A and B are recorded separately on user C's EyeReason server, they can be time shifted and relayed independently to user C's headset Based on user-specified buffering delay, first user A's recorded speech is played back at double speed to user C, after which user B's speech is similarly played back. User C can thus listen to both contributions in real time.




Buffered speech from two simultaneous speakers.

**Figure 5-5.**
Time-Multiplexing.

This process however is stopped when either user A or B falls silent. When this happens, the remaining buffer is first played back, after which user C can respond. Figure 5-5 explains this scenario graphically.

Initial experiences suggest that this time-multiplexing technique is most advantages in cases where two individuals simultaneously request the attention of a third, say for example to ask that person a question. It is less appropriate during group conversations,

where person A and B might both be interested in hearing each other's contributions. However, in such cases, both A and B may choose to use their pause button to buffer each other's speech for playback after they have finished speaking. We are currently investigating the implications of the above scenarios on comprehension, and on the conversational turn taking process in small groups.

## 5.3 Using EyeReason To Manage Human Group Communications: Attentive Office

The next step is to have *all* social interactions, including collocated ones, mediated by attention-aware systems. In office cubicle farms, where many users share the same workspace, problems of managing attention between co-workers are particularly acute. Our attentive cubicle system [33] addresses this problem by automatically mediating auditory and visual communications between co-workers on the basis of information about their socio-geometric relationships [10].

### 5.3.1 Attentive Office Cubicle

We designed an office cubicle that automatically mediates interruptions by co-workers. The Attentive Office Cubicle mediates both visual and auditory interactions between office workers by sensing whether they are candidate members of the same social group. The cubicle regulates visual interactions through the use of privacy glass, which can be rendered opaque or transparent upon detection of joint orientation. It regulates auditory interactions through the use of noise-cancelling headsets which upon co-orientation are programmed to become transparent to ambient sound.

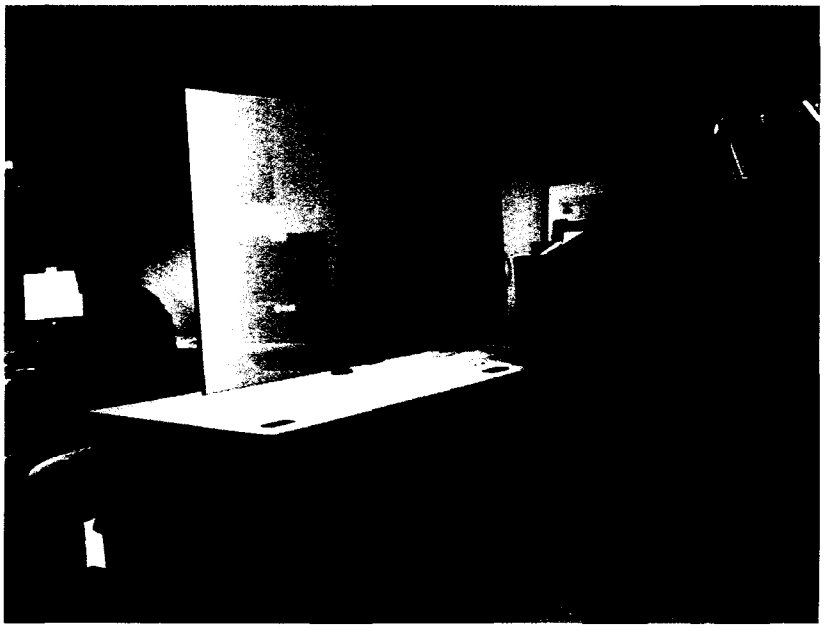


Figure 5-6a – In Opaque mode

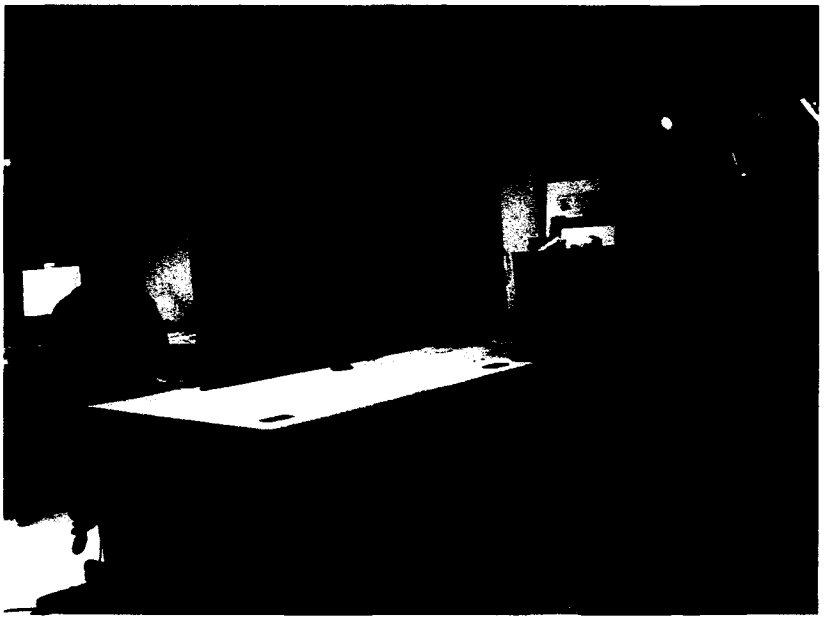


Figure 5-6b – In Transparent mode

**Figure 5-6.**
The Attentive Cubicle

Problems of managing attention between co-workers are particularly acute in cubicle farms, where many users share the same workspace. In order to avoid distraction, cubicle workers may opt to wear noise-cancelling headsets. Such headsets cancel out auditory distractions from co-workers and allow workers to focus better on their tasks. However, the use of noise-cancelling headsets places serious constraints on office collaborations as it reduces co-worker awareness of the environment. Our attentive cubicle system addresses this problem by automatically mediating auditory and visual communications between co-workers on the basis of information about their socio-geometrical relationships.

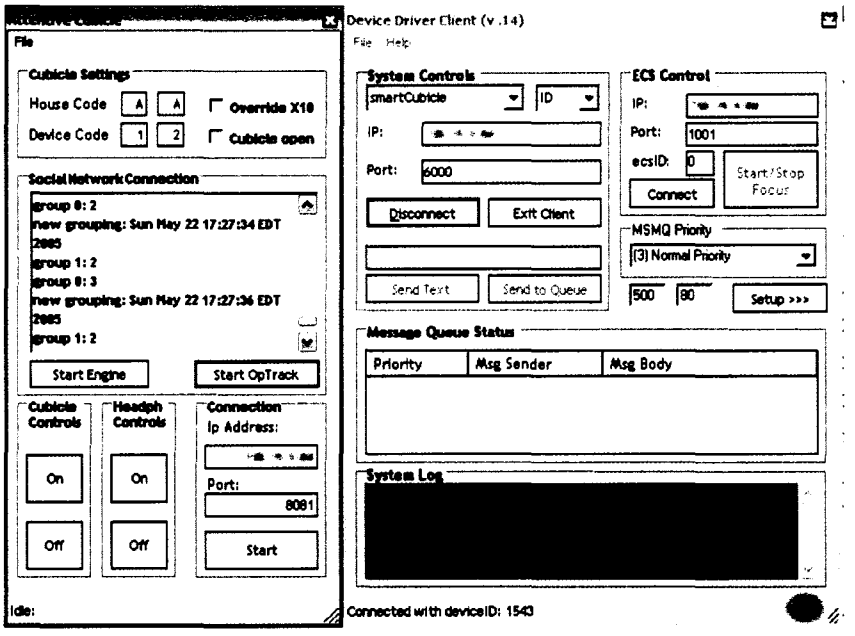
### 5.3.2 Implementation

Our prototype cubicle's walls were constructed using a special translucent material called Privacy Glass™ [57] (see Figure 3-3 on page 28). Privacy glass consists of a glass pane with an embedded layer of liquid crystals. When powered off, the crystals are aligned randomly, making the glass appear frosted and opaque (see Figure 5-6a on page 55). When a voltage is applied, the liquid crystals in the glass align, allowing light to go through the pane and rendering the glass transparent (see Figure 5-6b on page 55). When the privacy glass is opaque, cubicle workers cannot be seen by others and are not distracted by visual stimuli from outside. When the glass is transparent mode, the worker can interact visually with workers on the other side of his cubicle wall. We augmented the privacy glass with a contact microphone to allow our system to detect knocks by co-workers on the pane. These knocks inform the system of a request for attention of the opaque cubicle's occupant.

Each attentive cubicle worker wears a noise-cancelling Bose™ headset augmented with a fiducial marker and a microphone (see Figure 5-3[3] on page 49). Headsets are also augmented with a circuit that allows noise-cancellation to be switched on or off, and the signal from the microphone to be presented to the headset speakers. When the headset is turned off, wearers hear normally. When turned on, ambient sound is attenuated by –20 dB, thus allowing a wearer to work without auditory distractions.

Users within our office environment are tracked by our Social Geometry engine through overhead cameras mounted in the ceiling (see Figure 3-3 on page 28). For each tracked individual, the Social Geometry engine reports information about potential communication partners to that individual's EyeReason server [34] [50], as given by the ID of his or her fudicial marker. The EyeReason server controls the setting of the headset of the associated individual as well as the transparency of the privacy walls of a cubicle entered by that individual. Figure 5-7 shows the Device Driver Client for the Attentive Cubicle, called smartCubicle.

[3] Note that in the Attentive Cubicle, the headsets are not augmented with eye contact sensors. Instead they are replaced with the fiducial marker for tracking purposes.




The Device Driver Client for the Attentive Cubicle

**Figure 5-7.**
Screenshot of smartCubicle

### 5.3.3 Attentive Cubicle Scenario

The following scenario illustrates the use of the Attentive Cubicle system:

Sabrina is busy finishing a report. Sabrina has a tight deadline, as she needs to have the report filed by the end of the day. While Sabrina is trying to focus on her writing, her colleague Jeff, seated in the next cubicle, is discussing a design strategy with his co-worker, Laurie. All three individuals are wearing an attentive headset that is tracked by the Attentive Cubicle system. The cubicle recognizes that Laurie and Jeff are co-located and oriented towards each other, without any physical barriers between them. It reports each as a potential communication partner to the other person's EyeReason server. This causes their headphones to be set to transparent, allowing Jeff and Laurie to hear each other normally.

At the same time, the cubicle detects that co-worker Sabrina is not co-located with anyone, and is oriented towards her computer. Sabrina's EyeReason server is notified that there are no apparent communication candidates, causing it to engage noise cancellation and render her cubicle's privacy glass opaque. When Jeff and Laurie require Sabrina's assistance, Jeff makes a request for Sabrina's attention by knocking on the cubicle's privacy glass. The request is forwarded to Sabrina's EyeReason server, which informs the cubicle to consider removing the wall between the two individuals. It also causes Sabrina's noise cancellation to be turned off temporarily, allowing her to hear the request.

As Sabrina responds to the request, she orients herself to the source of the sound. The cubicle detects the co-orientation of Jeff and Sabrina. Sabrina's EyeReason server renders the privacy pane between Jeff and Sabrina translucent, allowing them to interact normally. After the conversation is completed, Jeff moves away from the cubicle wall, continuing his discussion with Laurie. Sabrina turns her attention back towards her computer system, causing the cubicle to conclude that Sabrina and Jeff are no longer candidate members of the same social group. Sabrina's EyeReason server responds by turning on noise-cancellation in Sabrina's headset, and by rendering the privacy glass of her cubicle opaque again.

The above scenario illustrates how entire rooms can be designed to balance social as well as privacy needs of co-workers in a dynamical fashion. It should be noted also that the scenario can also be applied to remote situations.

# Chapter 6

# Discussion

We now present a brief overview of some of our informal evaluations conducted with these prototype applications. These were designed to gather initial user experiences. More formal testing is required in order to evaluate the systems more rigorously. We end our discussions with a look at some future directions.

## 6.1 Experiences with AuraLamp and EyeTuner

Initial evaluations of the use of eye input for focus selection proved encouraging. According to Fono and Vertegaal [16], focus selection with the eyes is about twice as fast as with hotkeys or mouse. Verbal commands with AuraLamp were accurately communicated. The gaze activated speech lexicons of EyeReason properly ensured that

only the lamp that a user was paying attention to responded to his or her requests. We did notice that multiple ECS would sometimes interfere with one another, but when placed sufficiently far apart to prevent inadvertent reflections on the eye users appreciated the simplicity of selecting multiple AuraLamps using their eyes.

Within the context of EyeTuner, users were able to switch control between EyePliances with ease while playing and skipping media content. Users did note a lack of visual feedback on the detection of eye contact by an EyeTuner. In response we mounted a green LED on the ECS to turn on when the EyePliance gained focus. While evaluation of voice control has yet to be completed, switching of Bluetooth keyboard queries proved particularly promising, and may become an area for further research. However, users complained about a lack of feedback on their keyboarding actions. This led us to display their keystrokes on the EyePliance display.

## 6.2 Experiences with the Attentive Headphones

To appreciate user responses to the Attentive Headphones, and more generally, towards the idea of automated management of auditory attention, we informally tested our first prototype with laboratory colleagues. All participants had experience with the use of static noise-cancelling headsets in a lab-style environment.

Participants acknowledged that although they could better focus on their tasks, they were less aware of the activities around them. Participants found the peripheral ambient notification a useful feature, and reported that this allowed them suitable control over when and who interrupted them. However, some favoured a "call-waiting" tone or a cell phone ring-tone over the subtle disengagement of the headphones, as they would rather

not miss any audio currently playing in the headset. Participants reported that the headphones allowed them to continue their activities more seamlessly than what was possible using static noise-cancelling headsets.

## 6.3 Experiences with the Attentive Office

We operated the Attentive Cubicle system with four participants in our laboratory. Each pair of participants was asked to perform regular computing tasks, such as web-surfing and writing emails within their Attentive Cubicle while maintaining a conversation with his or her adjacent partner. EyeReason correctly identified when the pair was mutually oriented towards each other, and appropriately responded by switching the Privacy Glass to transparent. Three out of four respondents reported that the Attentive Cubicle reduced distraction levels, which suggests that active management of co-worker attention provides a promising approach. In the near future, we plan to evaluate our system more rigorously with larger numbers of participants and in real office cubicle environments.

## 6.4 Future Considerations

Throughout the process of designing Attentive User Interfaces, we came across many issues that have helped us identify outstanding research questions. Among the concepts explored, we found the metaphor of virtual windows of attention particularly inspiring. Whether in visual or auditory interactions with remote devices or people, users need to be supported by subtle cues that make up the *virtual windows* through which entities communicate with them. It is not sufficient to define such windows by the electronic channels through which interactions take place, because electronic channels do not delineate actual attention. By sensing user attention, devices may know when users are attending to them. By providing devices with a means of communicating their attention, users may know they are being attended to as well [6]. This allows users and devices to establish the negotiation of joint interest that is characteristic of multiparty human turn taking.

One of the technical problems we encountered is that of sensing attention for small or hidden devices. While physiological sensing technologies such as EEG and ECG monitoring may address these issues, they are potentially invasive. A second issue is the *identification* of users at a distance, rather than simple detection. While eye contact sensors may one day be able to perform iris scanning, there are privacy implications that must be considered. A third is that of prioritization of notifications. Can automated services be trusted to rank and prioritize information received according to human preference? This can prove particularly frustrating when such services "get it wrong". AUIs must be able to accurately translate sensor information into user intention, and to *understand* the importance and validity of messages to a user's current task.

One of the most pressing issues relating to the sensing technologies that have been presented is that of *privacy*. How do we safeguard privacy of the user when devices routinely sense, store and relay information about their identity, location, activities and communications with other people? These concerns may be the greatest barrier to the long-term success of ubiquitous computing. The key to alleviating some of these concerns lies in empowering users with choice and informed consent. Users gain control of the distribution of information pertinent to their needs and interests and have the potential to become more comfortable with context-sensitive ubiquitous technologies, when the method and form of dispersal of external information is placed directly into the hands of the user. [25]. Analogous to a good office receptionist, EyeReason provides a step in the direction of maintaining privacy by ensuring that any information retrieved from the user is not broadcast. However, EyeReason and EyePliances require an improved privacy securing architecture to ensure end-user trust and comfort. Work recently published by Hong and Landay on the Context Fabric (Confab) – an infrastructure aimed at simplifying the task of creating privacy sensitive ubicomp applications – may help in this regard [25].

# Chapter 7

# Summary & Conclusions

This research presented a framework for designing Attentive User Interfaces, the essential focus being the augmentation of user's attention. We explored this vision by developing EyePliances that communicate with a central EyeReason server. These interfaces negotiate interactions in ubiquitous environments where demands on a user's attention may exceed his or her mental capacity.

The purpose of EyeReason is to limit unnecessary interruptions, give remote interlocutors a sense of what activities they are intruding upon, and provide a facility to coordinate communications among EyePliances using a generalized model of user attention. The EyeReason architecture simplifies the process of augmenting a standard appliance with gaze and speech capability. By treating user attention as a limited resource, such interfaces reduce disruptive patterns of interruption.

Through reasoning about the Social Geometries formed by people's bodies during conversations, coarse-grained measures of body and head orientation allow us to determine the social engagement of participants within group meetings. In addition, embedding ubiquitous devices with attention sensors, such as eye contact sensors, devices can prioritize and manage their demands on user attention. This is turn allows users and devices to enter a turn taking process similar to that of human group conversation.

Initial experiences have proved promising and have highlighted the need to further investigate the role of AUI's in the design of HCI technologies. Within their design however, particular attention must be paid to ensure the proper security of personal information.

By designing virtual windows of attention between devices and users, communications in multiparty HCI may become more sociable as well as more efficient. AUIs, as such, may serve their ultimate goal: that of augmenting user attention.

# Bibliography


1. Apple Computers Inc. http://www.apple.com/airtunes, 2004.

2. Argyle, M. and Cook, M. *Gaze and Mutual Gaze*. Cambridge University Press, 1976.

3. ARToolKit. http://www.hitl.washington.edu/artoolkit/, 1994.

4. Basu, S. and Pentland, A., Smart Headphones. in *Extended Abstracts of CHI*, (Seattle, 2001), ACM Press.

5. Baudisch, P., Good, N. and Stewart, P., Focus Plus Context Screens: Combining Display Technology with Visualization Techniques. in *Proceedings of UIST'01*, (Orlando, Florida, USA, 2001), ACM Press, 31-40.

6. Bellotti, P., Back, M., Edwards, W.K., Grinter, R.E., Henderson, A. and Lopes, C., Making Sense of Sensing Systems: Five Questions for Designers and Researchers. in *Proceedings of CHI'02*, (Minneapolis, Minnesota, USA, 2002), ACM Press, 415-422.

7. Bolt, R.A. Conversing with Computers *Technology Review*, 1985, 34-43.

8. Carter, P. *Mies Van Der Rohe At Work.* Phaidon Press, 1999.

9. Cherry, C. Some Experiments on the Reception of Speech with One and with Two Ears. *Journal of the Acoustic Society of America, 25.* 975-979.

10. Danninger, M., Vertegaal, R., Siewiorek, D.P. and Mamuji, A., Using Social Geometry to Manage Interruptions and Co-Worker Attention in Office Environments. in *Proceedings of Graphics Interface '05*, (Victoria, British Columbia, Canada, 2005).

11. Deitz, P. and Yerazunis, W., Real-Time Audio Buffering for Telephone Applications. in *Proceedings of UIST 2001*, (New York, 2001), ACM Press, 193-194.

12. Duchowski, A. *Eye Tracking Methodology: Theory & Practice*. Springer-Verlag, Berlin, 2003.

13. Duncan, S.J. Some Signals and Rules for Taking Speaking Turns in Conversations. *Journal of Personality and Social Psychology, 23* (2). 286-288.

14. Einstein, G.O., McDaniel, M.A., Williford, C.L., Pagan, J.L. and Dismukes, R.K. Forgetting of Intentions in Demanding Situations Is Rapid. *Journal of Experimental Psychology:, 9* (3). 147–162.

15. Erickson, T. and Kellogg, W. Social Translucence: An approach to Designing Systems that Support Social Processes. *ACM Transaction on Computer-Human Interaction, 7* (1). 59-83.

16. Fono, D. and Vertegaal, R., EyeWindows: Evaluation of Eye-Controlled Zooming Windows for Focus Selection in *Proceedings of CHI '05*, (Portland, Oregon, USA 2005), ACM Press, 151-160.

17. Gibbs, W. Considerate Computing *Scientific American*, January 2005, 55-61.

18. Gillie, T. and Broadbent, D. What Makes Interruptions Disruptive? A Study of Length, Similarity and Complexity. *Psychological Research, 50*. 243-250.

19. Goldhaber, M. The Attention Economy and the Net, 1997.

20. Google Inc. http://www.google.ca/about.html, 1998.

21. Griffin, Z.M. and Bock, J.K. What the Eyes Say About Speaking. *Psychological Science, 11*. 274-279.

22. Hall, E.T. *The Hidden Dimension*. Doubleday, Garden City, NY, 1966.

23. Hall, E.T. *The Silent Language*. Doubleday, Garden City, NY, 1959.

24. Home Controls Inc. http://www.homecontrols.com/.

25. Hong, J.I. and Landay, J.A., An Architecture for Privacy-Sensitive Ubiquitous Computing in *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services '04*, (Boston, MA, USA, 2004), ACM Press, 177-189.

26. Horvitz, E., Jacobs, A. and Hovel, D., Attention-Sensitive Alerting. in *Proceedings of UAI'99*, (Stockholm, 1999), Morgan Kaufmann, 305-313.

27. Horvitz, E., Kadie, C.M., Paek, T. and Hovel, D. Models of Attention in Computing and Communications: From Principles to Applications *Communications of the ACM*, 2003, 52-59.

28. Hudson, S., Fogarty, J., Atkeson, C.G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J.C. and Yang, J., Predicting Human Interruptibility with Sensors: a Wizard of Oz Feasibility Study. in *Proceedings of the SIGCHI'03*, (Ft. Lauderdale, Florida, USA, 2003), ACM Press, 257-264.

29. Ishii, H. and Ullmer, B., Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. in *Proceedings of the SIGCHI'97*, (Atlanta, Georgia, USA, 1997), ACM Press, 234-241.

30. Jacob, R. *Eye Tracking in Advanced Interface Design.* Oxford University Press, Oxford, 1995.

31. Jacob, R.J.K. The Use of Eye Movements in Human- Computer Interaction Techniques. *ACM Transactions on Information Systems*, *9* (3). 152-169.

32. Maglio, P.P., Matlock, T., Campbell, C.S., Zhai, S. and Smith, B.A., Gaze and Speech in Attentive User Interfaces. in *Proceedings of the Third International Conference on Multimodal Interfaces*, (Beijing, China, 2000), Springer-Verlag, 1-7.

33. Mamuji, A., Vertegaal, R., Dickie, C., Sohn, C. and Danninger, M., Attentive Office Cubicles: Mediating Visual and Auditory Interactions Between Office Co-Workers. in *Video Proceedings of Ubicomp '04*, (Nottingham, England, UK, 2004).

34. Mamuji, A., Vertegaal, R., Shell, J.S., Pham, T. and Sohn, C., AuraLamp: Contextual Speech Recognition In An Eye Contact Sensing Light Appliance. in *Extended Abstracts of Ubicomp'03*, (2003).

35. Mamuji, A., Vertegaal, R., Sohn, C. and Cheng, D., Attentive Headphones: Augmenting Conversational Attention with a Real World TiVo®. in *Extended Abstracts of CHI'05*, (Portland, Oregon, USA, 2005), ACM Press.

36. McCrickard, D.S., Catrambone, R., Chewar, C.M. and Stasko, J.T. Establishing Tradeoffs that Leverage Attention for Utility: Empirically Evaluating Information Display in Notification Systems *Int. J. Hum.-Comput. Stud.*, *58* (5). 547-582.

37. McFarlane, D., Coordinating the Interruptions of People in Human-Computer Interaction. in *Proceedings of INTERACT'99*, (The Netherlands, 1999), IOS Press, 95-303.

38. McGonkie, G.W. and Rayner, K. The Span of Effective Stimulus During a Fixation in Reading. *Perception & Psychophysics*, *17*. 578-586.

39. Meier, R.L. *A Communications Theory of Urban Growth.* MIT Press, Boston, 1962.

40. Moran, T.P. and Dourish, P. Special Issue on Context-Aware Computing *Human Computer Interaction (HCI)*, 2001.

41. Morimoto , C.H., Koons, D., Amir, A. and Flickner, M. Pupil Detection and Tracking Using Multiple Light Sources. *Image and Vision Computing, 18.* 331-335.

42. Mueller, F. and Karau, M., Transparent Hearing. in *Extended Abstracts of CHI'02*, (Minneapolis, Minnesota, USA, 2002), 730 - 731.

43. Murphy, H. and Duchowski, A., Gaze-Contingent Level of Detail Rendering. in *EuroGraphics '01*, (Manchester, UK, 2001).

44. Nielsen, J. Noncommand User Interfaces *Communications of the ACM*, 1993, 83-99.

45. Oh, A., Fox, H., Van Kleek, M., Adler, A., Gajos, K., Morency, L.-P. and Darrell, T., Evaluating Look-to-Talk. in *Extended Abstracts of CHI 2002*, (Minneapolis, Minnesota, USA, 2002), 650-651.

46. Phidgets Inc. TextLCD Display, 2004.

47. Selker, T., Lockerd, A. and Martinez, J., Eye-R, A Glasses-Mounted Eye Motion Detection Interface. in *Extended Abstracts of CHI '01*, (Seattle, Washington, USA, 2001), ACM Press.

48. Shell, J.S., Selker, T. and Vertegaal, R. Interacting with Groups of Computers *Communications of the ACM*, March 2003, 40-46.

49. Shell, J.S., Vertegaal, R., Cheng, D., Skaburskis, A.W., Sohn, C., Stewart, A.J., Aoudeh, O. and Dickie, C., ECSGlasses and EyePliances: Using Attention to Open Sociable Windows of Interaction. in *Proceedings of ACM ETRA'04*, (San Antonio, Texas, USA, 2004), 93-100.

50. Shell, J.S., Vertegaal, R., Mamuji, A., Pham, T., Sohn, C. and Skaburskis, A., EyePliances and EyeReason: Using Attention to Drive Interactions with Ubiquitous Appliances. in *Extended Abstracts of Ubicomp'03*, (2003).

51. Shell, J.S., Vertegaal, R. and Skaburskis, A.W., EyePliances: Attention-Seeking Devices that Respond to Visual Attention. in *Extended Abstracts CHI '03*, (Ft. Lauderdale, Florida, USA, 2003), ACM Press, 770-771.

52. Short, J., Williams, E. and Christie, B. *The Social Psychology of Telecommunications*. Wiley, London, 1976.

53. Smith, D.C., Irby, C., Kimball, R., Verplank, W.L. and Harslem, E. Designing the Star user interface *BYTE*, April 1982, 653-661.

54. Stiefelhagen, R., Tracking Focus of Attention in Meetings. in *Proceedings of ICMI'02*, (2002).

55. Stiefelhagen, R. and Zhu, J., Head Orientation and Gaze Direction in Meetings. in *Proceedings of CHI'02*, (Minneapolis, Minnesota, USA, 2002), 858 - 859.

56. Suh, B., Woodruff, A., Rosenholtz, R. and Glass, A., Popout Prism: Adding Perceptual Principles to Overview+Detail Document Interfaces. in *Proceedings of CHI'02*, ( Minneapolis, Minnesota, USA, 2002), ACM Press, 251-258.

57. SwitchLight. SwitchLight Privacy Glass™ Specification, 2003.

58. TiVo® Inc. http://www.tivo.com, 2001.

59. Vertegaal, R., The GAZE Groupware System: Mediating Joint Attention in Multiparty Communication and Collaboration. in *Proceedings of SIGCHI'97*, (Pittsburg, Pennsylvania, USA, 1999), ACM Press, 294 - 301.

60. Vertegaal, R., Dickie, C., Sohn, C. and Flickner, M., Designing Attentive Cell Phones Using Wearable Eyecontact Sensors. in *Extended Abstracts of CHI '02*, (Minneapolis, Minnesota, USA, 2002), 646-647.

61. Vertegaal, R. and Ding, Y., Explaining Effects of Eye Gaze on Mediated Group Conversations: Amount or Synchronization? in *Proceedings of CSCW 2002*, (New Orleans, Louisiana, USA, 2002), ACM Press, 41-48.

62. Vertegaal, R., Mamuji, A., Sohn, C. and Cheng, D., Media EyePliances: Using Eye Tracking for Remote Control Focus Selection of Appliances in *In Extended Abstracts of CHI '05* (Portland, OR, USA 2005), ACM Press, 1861-1864

63. Vertegaal, R., Shell, J.S., Mamuji, A. and Chen, D. Designing for Augmented Attention: Towards a Framework for Attentive User Interfaces. *Special Issue on Attention-Aware Systems. Journal of Computers in Human Behaviour*

64. Vertegaal, R., Slagter, R., van der Veer, G. and Nijholt, A., Eyegaze Patterns in Conversations: There is More to Conversational Agents than Meets the Eyes. in *Proceedings of CHI'01*, (Seattle, Washington, USA, 2001), ACM Press, 301-308.

65. Vertegaal, R., van der Veer, G. and Vons, H., Effects of Gaze on Multiparty Mediated Communication. in *Proceedings of GI '00*, (Montreal, Quebec, Canada, 2000), ACM Press, 95-102.

66. Want, R., Fishkin, K.P., Gujar, A. and Harrison, B.L., Bridging Physical and Virtual Worlds with Electronic Tags. in *Proceedings of the SIGCHI '99*, (Pittsburgh, Pennsylvania, USA, 1999), ACM Press, 370-377.

67. Weiser, M. The Computer for the 21st Century *Scientific American*, September 1991, 94-110.

68. X10 Home Solutions. http://www.x10.com, 2003.

69. Zhai, S., Morimoto, C. and Ihde, S., Manual and Gaze Input Cascaded (MAGIC) Pointing. in *Proceedings of the SIGCHI'99*, (Pittsburgh, Pennsylvania, USA, 1999), 246-253.

# Appendix A

# TCP/IP Communication Protocol

## A.1 Handshaking Protocol

Note: All communication messages are sent using the ASCII format.

- Device connects to Server
- Establish connection to "EyeREASON server" on port "6001" (this is the initial port – port 6000 reserved for connection between device driver client and server)
- **Server:** "HelloDevice ¶ "
- **Device:**

| COMM_ID | DEVICE_ID |
|---|---|
| CONN | {0000,…,9999} |

| **Device** | **ID range** |
|---|---|
| smartCubicle | {1000,…,1999} |
| smartEmail | {2000,…,2999} |
| smartLight | {3000,…,3999} |
| smartHeadphones | {4000,…,4999} |
| smartTelevision | {5000,…,5999} |

- Server spawns the appropriate device driver for this device, and returns the port number that should be used to connect to this device driver.
- **Server:** "{6001,...,6999} ¶"
- Device then disconnects from server, and connects to the device driver using the new port number assigned.
- **Device:** "HelloMyDeviceDriver on Port {6001,...,6999>}¶"
- **Device Driver:** "ISeeYouDevice on Port {6001,...,6999}¶"
- **Device:** "PROPERTIES _ A _ B _ C _ D _ E _ F ¶"
- Device driver checks device properties, and if they match the expected properties for this device, then the device driver accepts the connection
- **Device Driver:** "ConnectionEstablished ¶"
- **Device:** "OK ¶"
- Otherwise, device driver tells the device that the properties are incorrect. Device then disconnects, and the device re-initiates the handshake again from the beginning.
- **Device Driver:** "PropertiesIncorrect ¶"
- **Device: "**QUIT ¶"
- **Device Driver:** "ClosingConnection ¶"

¶ = carriage return (CR) = ASCII 77

## A.2 Initialise Device Properties

PROP_ID = "<string>"

**A** *(Notifications):*

| PROP_ID | NOTIFICATION | END |
|---|---|---|
| NOTIF | D1 (256 bits mapped) | CR |

| | | |
|---|---|---|
| D1: | notify_disable | "0" |
| | notify_onlyaudio | "1" |
| | notify_onlyvideo | "2" |
| | notify_audiovideo | "3" |
| | notify_movement | "4" |

**B** *(Update Device State Properties in Device Driver):*

| PROP_ID | D1 | END |
|---|---|---|
| UPDATE | update | CR |

| | | |
|---|---|---|
| Update = | yes = | "1" |
| | No = | "0" |

**C** *(X-10 Console):*

| PROP_ID | CODE | END |
|---|---|---|
| X10 | ModuleCode | CR |

ModuleCode = "House ID + Code"

## A.3 Realtime Communication from Device Driver Client

COMM_ID = "<string>"

*Mode of notification:*

| COMM_ID | DEVICE_NAME | DEVICE_ID | AUDIO | VISUAL | MVMT | END |
|---|---|---|---|---|---|---|
| NOTIFCAP | <string> | {0000,...,0999} | "0, 1" | "0, 1" | "0, 1" | CR |

*Notification Channel Status:*

| COMM_ID | DEVICE_NAME | DEVICE_ID | AUDIO | VISUAL | END |
|---|---|---|---|---|---|
| NCHNL_STAT | <string> | {0000,...,0999} | "0, 1" | "0, 1" | CR |

*Activate Grammar:*

| COMM_ID | DEVICE_NAME | DEVICE_ID | END |
|---|---|---|---|
| GRAMM | <string> | {0000,...,0999} | CR |

*Deactivate Grammar:*

| COMM_ID | DEVICE_NAME | DEVICE_ID | END |
|---|---|---|---|
| END_GRAMM | <string> | {0000,...,0999} | CR |

*Activate Focus:*

| COMM_ID | DEVICE_NAME | DEVICE_ID | ECS_ID | END |
|---|---|---|---|---|
| FOCUS | <string> | {0000,...,0999} | {000,...,999} | CR |

*Deactivate Focus:*

| END_FOCUS | DEVICE_NAME | DEVICE_ID | ECS_ID | END |
|---|---|---|---|---|
| END_FOCUS | <string> | {0000,...,0999} | {000,...,999} | CR |

*Update Audio/Visual Level of the device:*

| COMM_ID | DEVICE_NAME | DEVICE ID | AUDIO | VISUAL | END |
|---|---|---|---|---|---|
| NCHNL_STAT | <string> | {0000,...,0999} | {0,...,100} | {0,...,100} | CR |

*Quit:*

| COMM_ID | DEVICE_NAME | DEVICE_ID | END |
|---|---|---|---|
| GONE | <string> | {0000,...,0999} | CR |

## A.4 Realtime Communication from EyeReason Server

COMM_ID = "<string>"

*Get Notification Channel Status:*

| COMM_ID | DEVICE_NAME | DEVICE_ID | END |
|---|---|---|---|
| NCHNL_STAT | <ID> | {0000,...,0999} | CR |

*Disconnect Device:*

| COMM_ID | DEVICE_NAME | DEVICE_ID | END |
|---|---|---|---|
| DISCONN | <ID> | {0000,...,0999} | CR |

*Send Message to Device:*

| COMM_ID | MESSAGE | PRIORITY | DEVICE_SRC | DEVICE_DEST | END |
|---|---|---|---|---|---|
| NOTIF | <string> | {1,...,7} | {0000,...,0999} | {0000,...,0999} | CR |

*Execute:*

| COMM_ID | EXEC_ID | NUM PARAM | PARAM LIST | END |
|---|---|---|---|---|
| EXEC | {500,..., 599} | {1,...,n} | <value> | CR |

# A.5 The EXECUTE Protocol

This protocol is used to execute remote commands to an EyePliance as directed by EyeReason. Each command is set at a specific ID.

→ To Device Driver Client
EXEC|deviceId|<execute id>|<# of params>|<parameter 1>|…|<parameter n>

→ From Device Driver Client:
- → EXEC_OK : Execution went OK!
- → EXEC_CERROR : Error in Execution Code (Does not exist)
- → EXEC_PERROR : Error in one of the parameters

Command Integer Reservations:

**Execute Functions {500,…,599}**

| FUNCTION | EXECUTE ID + PARAM | (SAMPLE) EXECUTION COMMAND |
|---|---|---|
| Play | 500 | EXEC\|{0000,…,0999}\|500\|0\|CR |
| Pause | 501 | EXEC\|{0000,…,0999}\|501\|0\|CR |
| Stop | 502 | EXEC\|{0000,…,0999}\|502\|0\|CR |
| Forward (N/A) | 503 | EXEC\|{0000,…,0999}\|503\|0\|CR |
| Rewind (N/A) | 504 | EXEC\|{0000,…,0999}\|504\|0\|CR |
| Volume Up | 505\|<% up> | EXEC\|{0000,…,0999}\|505\|1\|50\|CR |
| Volume Down | 506\|<% down> | EXEC\|{0000,…,0999}\|506\|1\|30\|CR |
| Volume (arbitrary) | 507\|<total %> | EXEC\|{0000,…,0999}\|507\|1\|75\|CR |
| Channel/Station Up | 508 | EXEC\|{0000,…,0999}\|508\|0\|CR |
| Channel/Station Down | 509 | EXEC\|{0000,…,0999}\|509\|0\|CR |
| Channel/Station (arbitrary) | 510\|<channel #> | EXEC\|{0000,…,0999}\|1\|32\|CR |