Maintaining Information Awareness in a Dynamic Environment: Assessing Animation as a Communication Mechanism

A Thesis
Presented to
The Academic Faculty

by

D. Scott McCrickard

In Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Georgia Institute of Technology
June 2000

Maintaining Information Awareness in a Dynamic Environment: Assessing Animation as a Communication Mechanism

Approved:

John T. Stasko, Chairman

Gregory D. Abowd

Amy S. Bruckman

Richard Catrambone

Mark Guzdial

Date Approved by Chairman

Acknowledgements

This thesis is the result of many years of effort and would not have been possible without the assistance of a great many people.

First, I would like to thank my advisor, John Stasko, whose steady guidance allowed me to develop my thesis at my own pace yet kept me on track to meet the many milestones of the PhD program. He seemed to know when to push me in new directions, when to encourage me to pursue a promising topic of interest, and when to step back and allow me to find my own way. He allowed me to make my own mistakes and to learn from them without showing impatience or frustration. His assistance has made this thesis possible, and I truly believe that his influence has helped me become a better researcher.

My other committee members helped me make my thesis well-rounded with broad appeal. Richard Catrambone provided invaluable assistance in a field in which I had little experience. His patient manner allowed me to expand my knowledge base, and his attention to detail made my final work much better than it otherwise would have been. Gregory Abowd, Amy Bruckman, and Mark Guzdial each provided important guidance and acted as a sounding board for my work, from the earliest stages of my research to the final preparation of this document.

I would like to thank Tom Rowan for sponsoring me at Oak Ridge National Laboratory for two summers. His expansive knowledge on a great many subjects was beneficial, and his willingness to share stories and ideas helped create an atmosphere ripe for creativity and innovation. The work from those summers provided the spark that led to this thesis work.

Thanks also to the various organizations that sponsored my work. In addition to support from the College of Computing and Graphics, Visualization, and Usability (GVU) Center at Georgia Tech, Laszlo Belady and the Mitsubishi Electric Research Laboratory provided early support for my work, and several student research grants from the USENIX Association allowed me to dedicate time to my research while helping to advance their cause.

I would like to acknowledge all of the other faculty at Georgia Tech with whom I had the pleasure of working and collaborating during my time at Georgia Tech. Their support and advice was helpful in every stage of the PhD process. In particular, Albert Badre constantly checked on my progress and always tried to keep his teaching assistant position open for me. I also thank him for nominating me for the College of Computing Graduate Teaching Assistant Award.

Thanks also to the College of Computing and GVU Center staff for their support of my efforts. They made my problems their own and did everything they could to help me solve them, and they managed to keep a positive attitude while doing so. Special thanks to Joan Morton for constantly coming to my rescue and to Peter Wan and Terry Countryman for bailing me out of problems late at night when it seemed like we were the only ones left in the building.

My fellow students at Georgia Tech helped keep me grounded during my years at Georgia Tech. Alex Zhao, with whom I collaborated on several papers, was always happy to give advice and seemed to know a little bit about everything related to computers. I am grateful for his efforts in developing the Agentk toolkit. David

Brogan always found a minute to listed to my ups and downs and is one of the finest people that I have ever known. Despite his University of Virginia affiliation, I hope we remain friends for years to come. Late-night Nintendo soccer matches with Jim Pitkow kept me sane during marathon work sessions. Thanks goes to Ron Metoyer and Steve Park for making sure that I kept up with happenings in the sports world. Despite his propensity to hang dirty laundry in the cube, Jason Ellis made life interesting with his sarcasm and witticisms. And who knows how many years ago I would have graduated if not for the many late night Krispy Kreme doughnut runs with Don Allison, who in the end kindly made sure that I was not the last person to graduate from the incoming class of 1992.

I feel lucky to be friends with numerous people in Atlanta outside of Georgia Tech that helped me escape from my cubicle-based life. In particular, the times I spent with members of my ALTA tennis teams were enjoyable. In particular, I would like to mention a few people. John Wilson may be the only person in the world to understand all of the rules of ALTA. His attention to detail ensured that everyone had fun and the team did well. Tony Lavorgna balanced humor and a drive to win during his time as team captain of both ALTA and USTA teams. Fellow computer geek David Goeckeler may have been one of the few people in this crowd to truly understand me. He was kind enough to allow me to work for him at his company to make some extra money. It was a great relief to know that Wally Tirado was always ready for a pickup tennis match, despite his Luddite ways that made him difficult to contact. Finally, Wick and Robin Carter were like family to me. Their love and care helped me keep a healthy perspective on life during my years in Atlanta.

Other non-tennis people that I would like to mention include Eva Regnier, one of the smartest and funniest people that I have ever met. Bobby Bodenheimer and Jeanne Larose seem destined for togetherness. They always seemed to be the life of the party no matter where we went. Andy Kogelnik was a good friend and occasional landlord when I would visit Lisa.

I was quite fortunate during my PhD career to be surrounded by numerous relatives, whose unwavering support helped me accomplish all that I did. My uncles, William Fowler of Knoxville and Arthur Fowler of Madisonville, and their families took great care of me when I worked in Tennessee and when I visited the family cabin there. It was a great comfort to have another uncle, Charles Fowler, and his family a short drive a way in Gainseville (GA) during most of my dissertation-writing days. Finally, it was a pleasure to have my cousin, Tavie Cobb, and her family down the road in Savannah. I enjoyed visiting them and absorbing the laid-back lifestyle of their historic southern city.

While at Georgia Tech, I was lucky enough to meet my fiancee, Lisa Wenner. Her adventurous nature and outgoing personality brightened my life from the first time that I saw her over my cubicle wall. She has managed to keep my life interesting despite my efforts to stay locked in my cube all the time. Her willingness to take care of all of the other things that arose while I was finishing my dissertation made my life much easier. I love her dearly, and I feel very fortunate that we will be spending the rest of our lives together.

Without the support of my parents, Don and Eleanor McCrickard, I never would have even thought to attempt so large an undertaking as a PhD degree. They encouraged me every step of the way, and they never lost faith in me even when times were tough. They supported me in every way possible, and I hope to be as good to my children as they were to me. I am very grateful to them.

Finally, I would like to acknowledge the role that my grandparents played in my development. My maternal grandmother, Elizabeth Thomas Fowler, exposed me to various experiences that enriched my life as a youth, most notably taking me on an extended tour of England when I was twelve. Her cabin on a lake in Tennessee remains in the family as a reminder of her energy and drive. My maternal grandfather, Joel Fort Fowler, died when I was young, but to this day I remember his kind and gentle manner. I am glad to have had the opportunity to know him. My paternal grandmother, Phyllis Alston McCrickard, always made sure to have a lemon meringue pie and a box of salt water taffy ready for me when I would come to visit. She tried her best to fatten me up a bit, and although she was unsuccessful I enjoyed her efforts. My paternal grandfather and only living grandparent, Holmah Austin McCrickard, remains a positive and energetic individual, always ready to trade stories and to share a laugh. He promised me that he would live to see me finish my thesis, and he kept his promise.

It is in memory of Elizabeth Fowler, Fort Fowler, and Phyllis McCrickard, and in honor of Holmah McCrickard, that I dedicate this thesis.

D.Scott McCrickard

Atlanta, GA

June 27, 2000

Contents

List of Tables

List of Figures

Summary

The Internet and World Wide Web provide a rapidly expanding pool of constantly changing information such as stock prices, news bulletins, sports scores, and weather data. Although people want to stay aware of changes in this information, their focus at the computer generally is on other activities, and they cannot afford to use large, attention-demanding displays. An alternative is the use of gradual, cyclic animations where only a small amount of screen space is required and changes to the information can be integrated smoothly in the next iteration. This thesis examines the tradeoffs between the awareness and distraction attributes of such animations. In particular, I have conducted empirical evaluations and observational studies of people using animated displays while performing typical daily tasks. Usage statistics, timed tasks, and questionnaires highlighted differences in performance based on the type of animation, characteristics of the user, and features of the information. These differences were used to develop a toolkit that facilitates the appropriate use of animated effects in desktop applications and to formulate a set of rules that dictates when and how animation can be effectively integrated into awareness applications.

CHAPTER 1
Introduction

The Internet and the World Wide Web can bring a plethora of information to people on their computer desktops. Much of the information available electronically now is dynamic: stock prices fluctuate, news bulletins arrive, email queues grow, alarms sound and the weather changes. In this thesis, *dynamic information* refers to semi-structured data that changes or is updated at frequent intervals. While many people feel overwhelmed by information, it is still critical for them to know about the information that is important to them. For instance, people care when an important new software release occurs, when email arrives from their parents, when traffic is horrible on the interstate going home, when their file is caught behind five 20-megabyte jobs in the printer queue, and when a ball game is occurring that involves their favorite team. Clearly, the availability of this type of information *can* be good as long as it is desired information that is presented in an appropriate and non-intrusive manner.

With most information, especially static or infrequently-changing information, people seek to answer a question or make a decision. Consequently, they examine an interface, come to some conclusion, and move on. But with dynamic information, people's tasks more closely align to *awareness*, a process of maintaining knowledge about the state of and changes to a body of information. The desire to maintain awareness can occur on many levels that involve a wide range of priorities for information, bordered on one extreme by the monitoring of information for specific changes that will potentially change one's current actions and on the other extreme by a mild interest for information that will not affect one's current course of action.

As it is a new field, relatively few techniques have been explored for maintaining awareness of the contents of dynamic information sources. Current computer display tools generate innovative and complex visualizations, yet they can demand a person's full attention or can distract people from other daily tasks and as such are not well-suited for communicating information over a long period of time. While dynamic information is useful to a person throughout their daily activities, it often is complementary to other tasks such as entering data, editing a document, programming code, or writing. People want to maintain awareness of information, but they do not want this awareness to overwhelm their other work activities.

As such, it is advisable to communicate information in the periphery to allow people to focus on a primary task. While there are various means to integrate information in the environment, practical concerns motivated me to focus on alternatives suitable for the computer desktop. Since the majority of the screen space is dominated by tools necessary to support the primary task, peripheral displays must be fairly small. People use a variety of tools that fall into the category of peripheral displays, including clocks, load monitors, and email alert tools. Thus, it is reasonable to expect that people would be willing to use a peripheral tool for raising awareness of dynamic information.

Given that the information should be communicated in the periphery, what mechanisms can be used to facilitate it? One solution is the use of animation as a visual tool in maintaining awareness. What is animation? Baecker and Small describe it as "sequences of static images changing rapidly enough to create the illusion of a continuously changing picture" [3]. It has been used to generate emotion, provide entertainment, and supply information.

In general, animation refers to a visual change in an image or text, including changes in position, color, size, texture, or content. In this thesis, *animation* typically refers to constant and cyclic changes in information. When used in this way, animation can show a large amount of information in a small space in a hands-off manner. Tools such as stock tickers, sports wires, and banner advertisements use this type of animation to convey information, but such tools often have been criticized as being too distracting. It is my expectation that much of the distraction stems from the nature of the animations and the tasks for which they have been used. People may be willing to tolerate and even appreciate animation, but only if its benefits outweigh any potential drawbacks. I have been studying different types of animations to better understand their capabilities and utility for presenting dynamic information.

The focus of this work can be captured in the following thesis statement:

> Constant and cyclic animations, when used in appropriate situations, can assist an individual in maintaining awareness of dynamic information.

This research contributes to the field of human-computer interaction by exploring the role of animation in maintaining awareness of dynamic information. In performing this research, I identified ways in which characteristics of peripheral displays can affect performance on both primary and peripheral tasks via empirical evaluations. I created a user interface toolkit that simplifies the inclusion of gradual, cyclic animated effects in peripheral awareness applications. I categorized users based on personal and situational qualities that then can be associated with animation preferences. In so doing, this thesis extends the understanding of the usefulness of peripheral animated displays.

The early stages of this thesis work were dedicated to exploring different ways to communicate information in the periphery. I built an information monitoring and display system that presented information using a variety of different communication mechanisms. A user study examined the ways in which people used the tool and analyzed which were the most effective. Chapter 3 describes this background work and explains how it led to the selection of animation as an awareness support technique for further study.

Having chosen to examine animated peripheral displays, the next step was to demonstrate their viability as a technique for helping users maintain awareness while completing other tasks. I designed a tool that presented information using a variety of types of animated displays and conducted a pilot study to survey if and how the tool would be used in a situation of fairly significant importance. Chapter 4 describes the pilot study and the lessons learned from it.

While the pilot study answered many questions, several new concerns were raised about the relative effectiveness and potential distraction of peripheral animated displays. To address these concerns, I conducted several empirical evaluations that studied the usefulness of animation in peripheral awareness displays. The evaluations examined qualities such as the type, size, and speed of the animated display. Chapter 5 provides details and results from these evaluations.

In building the tools and environments used in my evaluations, I noticed a lack of support in existing toolkits for the type of animation that proved to be useful in maintaining awareness. To address this need, I developed a toolkit that facilitates the programming of animated effects for awareness situations. Chapter 6 introduces this toolkit and outlines how it enables programmers to include animated effects into their programs quickly and easily.

The data from the empirical evaluations indicated that performance differences are related to characteristics of the animated display, but further research was needed to understand whether personal characteristics may also affect performance. I expanded the tool from the pilot study to include additional informational and display options and then conducted an observational study that asked in-depth questions and monitored users' interactions to gain a deeper understanding of how users are similar and how they differ. The study resulted in a set of user profiles that could be used to predict animation preferences. Chapter 7 gives details about the observational study.

Finally, Chapter 8 provides some conclusions, describes potential future work, and outlines the contributions of this thesis.

CHAPTER 2
Related Work

This chapter examines previous work that is relevant to this thesis. It is divided into categories that correspond to the areas in which this thesis makes contributions. Section 2.1 helps to frame the awareness and monitoring problem as defined in the introduction. The section takes a broad view of the needs of users and the types of information that can be communicated. Section 2.2 examines different communication alternatives that can be used in monitoring and awareness, with Section 2.3 focusing on animation as a communication alternative. Section 2.4 explores different ways of including animation in user interfaces and discusses the need for specific techniques for using animation to help raise informational awareness. Section 2.5 examines techniques for evaluating awareness and animation and compares them to the techniques used in this thesis.

2.1 General approaches to awareness and monitoring

This section presents research that motivates the need for the communication alternatives described in this thesis. First, I examine the generation, collection, and processing of information by software agents, autonomous programs that run constantly and need to make users aware of their findings. Next, I examine motivating activities related to this type of information processing, focusing on awareness and monitoring activities. In so doing, this section helps to develop an understanding of why it is essential to identify appropriate communication methods for dynamic information resources, the identification of these methods being central to this thesis.

2.1.1 Motivating research

One driving force necessitating awareness interface techniques is the development of software agents. A software agent is a system that performs tasks in an autonomous manner in order to satisfy the needs of its users. Of particular relevance are agents that constantly collect and process information, then need to communicate the information to the user. Popular Web-based agents include Letisia, a system that observes users' browsing choices, then looks ahead to make recommendations [43]; Firefly, which uses collaborative learning to match users with information of interest [68]; and PointCast and the many other applications (BackWeb, AfterDark) that repeatedly download news, sports, and weather information and displays it to the user [63, 2, 1]. Often the information obtained by these agents is not of critical importance, but it is expected that overall productivity can be improved by being aware of this information.

Of relevance to this work is the flexibility to communicate information collected by agents in the appropriate manner. An example of this is Apple Data Detectors, a series of tools that scan selected text and offer potential actions based on the structure of the text (see [57]). For example, a user can open or bookmark URLs, send email, read newsgroups, or retrieve files from FTP sites. Other such agents can be developed by Agentsheets, a spreadsheet-like visual programming application for creating information-triggered agents [65]. Examples of agents created by Agentsheets range from educational simulations of bridge building and electricity to information processing systems that interpolate temperature over a large region based on a few points.

The contribution of the work in this thesis is not in the creation and understanding of agent technologies, but in the development of awareness techniques for the type of information collected by these agents. As information sources found on the World Wide Web and similar repositories become broader in scope and more frequently updated, the need for appropriate communication techniques will become even more pronounced.

2.1.2 Awareness and monitoring

In considering motivating user tasks that have necessitated the development of agents, two closely-related tasks that are often (incorrectly) used interchangeably are information awareness and information monitoring. As such, it is important to examine the similarities and differences between them. Awareness is a more passive task whereby an individual wants to maintain knowledge of a changing resource without interrupting a primary task. Real world devices that help support awareness include speedometers and clocks. In monitoring, an individual still wants to know about changes, but specific changes typically trigger the interruption of the current task to undertake a different task. Real world devices that help support monitoring include alarm clocks, security cameras, and beepers.

Monitoring is probably the more widely addressed and widely understood of the two, with numerous examples of monitoring systems available. Jonathan Cohen provides a support structure for monitoring based on audio alerts [16]. More recently, work by Lyn Bartram has explored the perceptual properties of motion as it proves useful in communicating information [5]. In both examples, the communication technique is designed to draw attention away from the current task to some other task, useful in a monitoring situation.

Awareness typically involves a desire to acquire or maintain knowledge about changing information without distracting the user from other tasks. In studying the communication properties of animated effects, Paul Maglio is careful to focus only on information gained on the periphery, while at the same time measuring the distraction peripheral displays may cause in performing a primary task of editing a document [45]. However, studies like these may create an unrealistic testing environment given that the distinction between monitoring and awareness is not firm. For example, radios, cuckoo clocks, and television could be viewed as either monitoring devices or awareness devices. Cadiz's Awareness Monitor system semantically and functionally emphasizes the blending of these concepts, using a series of effects (including tickers) to address both monitoring and awareness issues [11]. Monitoring and awareness tasks are often closely related. For example, individuals in general may want to be aware of a stock quote, but during critical times they may want to monitor it. Similarly, when keeping track of the score of a ball game, in general it may be desirable to be aware that one team is far ahead of the other or that it is early in the game, but if the score remains close near the end of the game a viewer might want to focus attention on the game. While the main focus of this thesis is on awareness, at times the tasks I chose to examine cross into what many would consider to be monitoring.

2.2 Communication alternatives

Several options have been explored in communicating information to aid in awareness and monitoring.

Recent work has focused on non-desktop methods for maintaining awareness. Hiroshi Ishii's Tangible Media Group at MIT coupled information with everyday physical objects to communicate information at the periphery using light, sound, air flow, and water movement [35]. Scott Hudson at CMU adopted a similar approach, but he focused on techniques that do not rely on the projection of light in order to lessen the possibility for distraction [30]. Background audio also has been used to assist in monitoring and awareness in Mynatt's Audio Aura and Cohen's auditory display system [56, 16]. Though many of the techniques described in this thesis are applicable to non-desktop systems, to simplify testing and increase the potential user population this thesis focuses on displays in the periphery of the desktop.

Desktop communication often focuses on graphical displays of information. A well-designed graphical display can communicate information using a small amount of space. Graphical communication mechanisms have been making their way onto computer desktops as the speed and processing capabilities have made them viable. The rules and guidelines for the use of graphics established and explained by Edward Tufte [77] suggest that communication using graphics can be advantageous because more information can be communicated in a smaller space than with text alone, and because graphical images can often be processed with less cognitive effort than can textual ones.

In cases where the information to be communicated is graphical, graphical images should be used to convey the information. This would allow the information to be conveyed with minimal changes to its form, minimizing distortion of its meaning. For example, a Web site that consists largely of images might be best represented by showing parts or all of these images when they change. This technique was used in Helfman's passive Web browser that used pictures from a proxy that had been loaded by other users [31, 32]. A similar technique was used in Andruid Kerne's Collage Machine system [39] and Brown and McCrickard's CWIC system [9]. However, the drawback to these systems is that images from Web sites are typically large and require a fair amount of precious screen space. These systems overcome that either by requiring the user to read mail [31], visit a Web page [9, 39], allow a screen saver to run [9, 63], or view the screen wallpaper [9] to view the images. Still, this does not constantly raise awareness while allowing the user to perform other tasks using the same screen real estate.

One way to raise awareness in a small space is with icons. An *icon* is a small picture used on buttons and other widgets to communicate information about functionality or contents. For example, the"mailbox flag down" icon on many email monitoring programs programs (for example, the Unix xbiff program [25]) indicates that there are no unread messages, while the "flag up" icon indicates that there are messages. Of particular relevance to this work are *animated icons*, small images that change to represent actions [4]. For example, a drawing action may be difficult to represent in a graphics palette with a static icon, but an animated icon can show each of the steps in the action. Empirical evaluations have shown that the use of animated icons to represent complex word processing tasks results in improved performance [6]. However, much of the information that needs to be communicated is textual, and it is acknowledged to be difficult to generate icons for arbitrary textual information [4].

As such, the focus of this work turned to techniques that would support textual information. Of course, the earliest information monitoring and awareness tools, which ran on text-only displays, had to use text to communicate information. The biff email tool interrupts the user's current activity by printing the header and first few lines of a new email message in a terminal window [78]. In a similar manner, the wall tool displays information such as machine shutdown times on the terminals of all currently logged-in users [80]. In windowing systems, tools such as plan demand immediate attention by popping up windows containing textual messages about events on one's personal calendar [18]. These tools fall more in the category of monitoring tools, with the goal being to interrupt the current task of the user with one of higher immediate importance.

While this is an important category, it is beneficial to consider less intrusive methods. One such method is smooth, constant animation. It is widely acknowledged that this type of animation can become less intrusive over time, and the use of animation has interesting potential in maintaining awareness. The next section looks at ways in which animation has been used as a communication mechanism.

2.3 Communicating using animation

Animation has been used in various information communication situations. Baecker and Small listed several uses for animation, including identification, transition, demonstration, history, guidance, and feedback [3]. Based in part on this list, Stasko surveyed existing utilizations of animation in the interface and identified design principles for animation in interfaces, including appropriateness, smoothness, duration, control, and moderation [73]. Some examples of the use of animation include the percent-done indicators for providing feedback [54], animation of icons for demonstrating corresponding physical actions (a paintbrush with painting motions, a pencil with drawing motions) [4], algorithm animation to assist in communicating changes in state [72], and the use of motion to attract attention to or show relationships between objects [5]. One use conspicuously absent from the Baecker and Small list is the application of animation to better utilize screen space. For example, cone trees [66] use animation to show more information (in this case about hierarchies) than would otherwise be possible in a given space. I feel that this would provide a distinct advantage in secondary awareness tasks where space is at a premium.

Studies have shown that animation can be useful in certain decision-making situations [27]. In these studies, participants completed a home selection problem and a physics problem, and they were evaluated based on completion time, accuracy, ease of use, and enjoyment. Participants performed better with smooth, realistic, interactive animations compared to abstract representations and abrupt changes. In these and similar situations, the animation often has a significant visual (often physical) component: either it deals with a physical object, or it has a commonly-associated visual representation. It is unclear from these results whether the use of animation will be effective in broader, more abstract problem domains such as information communication.

One such abstract problem domain is algorithm learning and understanding. While Stasko's path-transition paradigm for algorithm animation was developed with the expectation that smooth animation rather than sudden changes would help to communicate the information in a more understandable manner [72], initial empirical research on its benefits showed disappointing results [74, 10]. However, other work has been more promising, particularly in hands-on situations where users have significant control over the algorithm animation. One study showed that the use of animation was beneficial when students were permitted to enter their own data as input [42]. Another study showed that algorithm animations proved particularly useful in interactive learning situations [38]. However, a high degree of interactivity cannot be expected in the secondary monitoring and awareness situations explored in this thesis.

The communication abilities of animation in other learning situations have been explored as well. A study compared the use of animated demonstrations, written text, and a narrated animation for teaching users how to operate a particular graphical interface [61, 60]. Although participants completed the initial training task more quickly when using animation, they completed the testing task more slowly. Furthermore, after a week, participants who used the animation performed more poorly on the task. This work showed that animation can be effective in the short term for learning, but often the learned information is not retained in the long term, perhaps because participants merely mimicked the behavior they saw in the animation rather than attempting to learn the proper behavior. It is worth noting, however, that the learning tasks in this situation are quite different from the awareness tasks explored in this thesis. It is expected that in awareness tasks, users will use the animation not to learn new information, but merely to recognize that a change has occurred. They will use more suitable interfaces to learn the information if desired.

While the prior work described in this section suggests that animation may be useful in other domains, it is difficult to make direct connections between the results described here and those expected for peripheral awareness tasks. The animated displays described in this section are typically either the primary focus of the user or are designed to draw the attention of the user to the animated information. Furthermore, one common conclusion is that animations are more effective when they are highly interactive, an undesirable trait for secondary, peripheral displays. Most importantly, the tasks that the users are trying to accomplish (learning and decision making) typically are more cognitively demanding than the monitoring and awareness tasks considered in this thesis. Little work has been done in understanding the usefulness of continuous, long-term animation for secondary awareness tasks.

Recent work has seen attempts to include animation in awareness devices, but many of these devices are dismissed as being too distracting, and little testing has been done to understand their potential utility. Constant, cyclic animations such as those studied in this thesis can be seen in stock quote displays, advertisements, and sports scores used by many programs and on many Web sites. For example, Yahoo released a stock ticker in 1996 that has experienced several iterations and is still available (see [81]). Pointcast is perhaps most widely remembered as a user of animation, employing horizontal scrolling effects and other effects in their screensaver and standalone application [63]. Numerous Web sites, including the Golf Channel (www.thegolfchannel.com), the Atlanta Visitor's Bureau (www.avcb.com), and the Georgia Tech College of Computing (www.cc.gatech.edu) at one time used tickering effects to communicate information, but each has now abandoned the effect, no doubt after complaints from users. Rather than abandoning the use of animation entirely, this thesis attempts to understand how animation can be used effectively and when potential users would be more likely to utilize it.

2.4 Supporting animation in interfaces

Animation has long been seen as a potentially important part of information communication. In the 1960s, the SketchPad project used simple animation and speculated on ways in which animation would be used in the future [76]. The Xerox Alto and Star introduced animation to the computer desktop [70]. Lassiter bridged the gap by using Disney-like effects like anticipation, slow-in/slow-out, and squashing in the user interface [41]. Chang and Ungar included many of these effects in a user interface toolkit to simplify their inclusion in interfaces [14].

Several widget sets have been developed to allow the programmer to include animation in the interface. The Artkit toolkit allows programmers to create transition objects that describe how an object will move [34]. A reference to the transition object is then added to a graphical object to create the animation, and Disney-like effects can be associated with the animated effect. Another toolkit, Amulet, was extended to include support for animation [55]. Animation can be attached to any value of any object, including position, color, or visibility. These and similar toolkits provide a great deal of power, yet they often can require significant effort by the programmer to achieve a desired result.

The widget set developed as part of this work does not require the user to specify the details of the animation. Instead, in only a few lines of code, a programmer can specify variables, define animation behavior, and start a cyclic, repetitive animation that automatically updates when variables are changed by the program. This same behavior would be much more difficult with most other toolkits. At the same time, the animated widgets are structured such that new animated widgets can be added with significant code reuse.

2.5 Evaluating animation

To separate this work from ad-hoc uses of animation, it is necessary to consider why animation would be useful in maintaining awareness. My plan was to accomplish this using two types of evaluations: empirical evaluations and observational studies. In this work, an *empirical evaluation* refers to a study in a controlled environment using compensated participants performing a well-defined set of tasks. An *observational study* is a less-controlled evaluation that places an interface in the hands of potential users and monitors their behavior over a long period of time. While the results do not have the statistical robustness of empirical evaluations, they capture users' opinions in more realistic working environments. As both types of evaluation were used during the development of my toolkit, this section will examine examples of each that are related to my thesis work.

2.5.1 Empirical evaluations

Empirical evaluations allow experimenters to better understand how certain factors can affect performance on various tasks. Since most variables in empirical evaluations are fixed, it is possible to make strong conclusions about the effects from varying a few select variables.

Some of the earliest evaluations of animated displays examined the perceptibility and readability of rapid serial visual presentations (RSVPs) of letters, strings, and words. Foster studied the readability of sentences that were presented a word at a time in a single visual location using motion picture film [23]. He found that graduate student participants could correctly identify about four out of six words in a sentence when presented at 62.5 milliseconds per word. Juola extended this word to also consider comprehension of information on computer screens presented as RSVPs and in multi-line paragraph format [36]. He found that comprehension was comparable in the two presentation modes.

The first studies of smooth animated effects were performed in the early 1980s. Duchnicky and Kolers performed a series of experiments examining the readability of text scrolled on visual display terminals as a function of window size [19]. They found that larger displays typically led to faster performance on reading tasks. A study led by Michael Granaas found that larger jumps (four to ten characters) led to better comprehension than smaller jumps (one to two characters) [28]. Kang and Muter compared a tickering effect to a non-animated "blast" effect where the information was changed without a gradual animation [37]. They found no difference in comprehension for a display reading task.

The previously mentioned evaluations all examined the reading of animated displays as the sole task of the participant. Only recently and concurrently with the work in this thesis has the usefulness of animation in maintaining awareness and in monitoring events been explored. Paul Maglio performed a series of experiments to examine the tradeoffs in displaying peripheral information [45]. Participants performed a series of primary tasks where they were asked to edit a document. Simultaneously, a continually scrolling, start-and-stop scrolling, or fading display would show information. He concluded that continually scrolling displays are more distracting than displays that start and stop, but information in both is remembered equally well. Scrolling direction does not seem to affect performance, and additional cues that are auditory have a more negative impact than additional visual cues.

While these results are interesting, they provide a narrow view of the usefulness and impact that peripheral animated displays can create. The work described in this thesis examines effects on a browsing task, a cognitively less demanding primary task that better matches the expected usage pattern noted in usage studies of animated displays (see [46, 22, 62, 52] and Section 5.3). These studies showed that many users would not use animated displays at all times, but rather only during selected low-effort primary activities. Also, the work in this thesis considers not only awareness activities but also monitoring activities that require participants to perform a simple task when they notice certain changes in information. This combination of activities again seems to better match the different types of activities that users would perform in the real world. Even though the Maglio work includes visual and auditory cues that are designed to draw attention to changes, this work does not specifically test whether participants notice these changes, only whether they are distracted by the additional cues.

2.5.2 Observational studies

While empirical evaluations provide evidence that a task can be done with a given interface in certain situations, one criticism of them is that they often take place in unrealistic environments with contrived tasks. Observational studies are important in demonstrating that potential users are willing to use the interface technique to accomplish real-world tasks.

Several early observational studies that I performed contributed to the direction of this thesis. The Irwin study examined how a set of communication techniques (audio, graphical encodings, animation) impacted awareness for four users of constantly changing resources such as email, Usenet news, weather, and certain Web sites [46]. The development of the tool and the study conducted to examine its use is described in Section 3.1. The tkscore study focused on animated and non-animated displays of ball scores during the single-elimination NCAA basketball tournament [52]. The reactions of over twenty users to various aspects of the different displays is described in Section 4.1.

Other similar work at the Georgia Tech GVU Center that is currently under way is an observational study by Alex Zhao of his What's Happening (WH) system. Much like the evaluation of the Irwin system, the WH evaluation examines how a set of communication techniques can help users maintain awareness of information, in this case information about local events that could help build a sense of community. Initial results of the study can be found in [82].

Numerous observational studies conducted elsewhere highlight guidelines for obtaining useful results. Notable among these is work such as the Bellcore VideoWindow study [21]. The VideoWindow provided an audio and a video link between two sites, then used cameras to observe how users interacted when using the VideoWindow as compared to face-to-face interactions. The large amount of data collected allowed the experimenters to make interesting conclusions about the reluctance of participants to use this new technology. By focusing on a limited set of related communication techniques, the authors were able to understand utility and establish guidelines for the techniques.

While the studies described previously in this section have examined a variety of communication mechanisms, fewer studies have focused on a single technique as in the case of this thesis, constant and cyclic animation. Perhaps the work most similar to the observational study described in this thesis came from the Elvin project [22, 62]. This project included a system called Tickertape that collects information from Web sites and newsgroups, then allows users to send and read messages that are displayed using a single line scrolling message window. An observational study looked at twenty Tickertape users who were interviewed in semi-structured sessions that often digressed into conversations. The result was a categorization of users based on the tasks that Tickertape helped to support: work, social activities, leisure, or news watching. The existence of categories were supported by sample dialogues and quotes from users. It was not clear if there were specific characteristics of users that made them more likely to fall into a certain category.

One problem with many of these studies is that they try to evaluate too many different techniques. For example, our experiences in evaluating Irwin showed that people found the tool informative and interesting, but it was unclear if this was because of the graphical encodings, the audio cues, the textual information, or a combination of these factors. Even the Tickertape study, which involved a system that used a single communication technique, was more of an exploratory study examining potential uses of an awareness system rather than an evaluation of the effectiveness of an animated display for different users. The goal of this thesis is to explore the effectiveness of a single class of techniques and to understand the different ways that potential users might react to animated techniques. In particular, the observational study categorizes user behaviors based on shared characteristics of users with similar preferences. See Chapter 7 for a full report of the study.

CHAPTER 3
Background and Motivation

Numerous people spend long hours each day at a computer. Their primary tasks vary significantly from composing documents to writing code to entering data to searching for and reading information. While one would expect that the primary task would dominate the machine, few of these tasks require the full resources of the system. Of relevance to this thesis, users have proven to be willing to sacrifice the periphery of the screen for informational displays not necessarily related to the primary task, including clocks, load monitors, task bars, and more.

What types of display techniques can be used effectively in the periphery? Graphics can encode large amounts of information in a small space. Auditory cues can inform the user without occupying any screen space. Animation can cycle through information, informing the user without requiring physical interactions.

What types of tasks are best suited for this type of peripheral display? Various tools support awareness and monitoring, though the ever increasing amount of information on the Web presents a new domain to test their effectiveness. Peripheral displays could provide a quick way to perform browsing, searching, and reading tasks without having to raise a window specifically for the task. However, the problem remains of matching display techniques to tasks in a manner that people enjoy using and can use effectively.

This chapter presents Irwin, a peripheral tool that employs the communication mechanisms described above in a peripheral display. A case study examined how Irwin users made use of the tool, and the relative merits of peripheral displays for a variety of tasks are discussed.

3.1 Irwin

This section explores display techniques used in peripheral displays to examine if and how people will use them, and how they can be adapted and improved to optimize the use of the available space. I created a tool called Irwin to help understand these issues. It was designed to occupy a small amount of screen space such that it could be positioned in the periphery and run constantly to keep the user informed of changes to selected information resources. Irwin uses graphical encodings, text, auditory cues, and animation in presenting information to the user. A case study examined how four people used Irwin in their workplaces over a five-month period.

3.1.1 Irwin description

Irwin [51] monitors Internet information resources and alerts the user of updates and modifications. Irwin consists of a set of hypertools–small reusable programs that can run simultaneously and share information. The central tool in Irwin handles the visualization and user interactions, while the remaining tools process the information from each resource and update the visualization tool when important changes occur.

The information resources monitored by Irwin can include email folders, Usenet newsgroups, Web pages, and weather data. The email and newsgroup hypertools monitor the messages in a folder or group, alerting the user when new messages arrive. The Web tool summarizes headers, lists, and hypertext links on a Web page, allowing the user to monitor news wires and hotlists. The weather tool monitors the weather conditions and forecast for a given city. Each results in a list of information that is then displayed by a variety of communication techniques.

Irwin uses multiple communication techniques to convey an overview of each resource as well as details about a selected resource: a set of animated icons, an auditory cue, a navigation bar, and several textual views. In the set of icons, the state of each resource is given by an icon. As the resource changes, the appearance of the icon changes and an auditory cue is played. If user selects an icon, the other views are updated to show information about the corresponding resource. A navigation bar shows a syntactic encoding of the resource contents and can be used like a scrollbar to scroll through the messages displayed in the textual views. Users can configure the orientation and placement of the icons with respect to the other views and can even choose to hide the other views until some event happens, e.g. an icon is clicked. Figure 1 shows the Irwin interface; the remainder of this section describes each communication technique that it uses.

3.1.1.1 Animated icons

Icons are used in interfaces because they provide a universal representation in a small amount of space. In windows-based environments, they can both provide information and invite the user to click to obtain more information. Studies have shown that changing the appearance of icons can convey additional information about their use; for example, animated icons [4] convey more information about the functionality of tools in a tool palette than a static image.

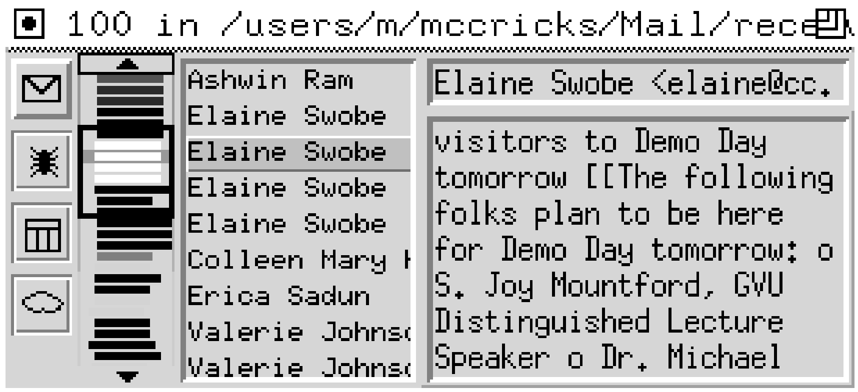

Figure 1: The Irwin user interface. The resources (email, a Web page, a Usenet newsgroup, and the local weather) are represented by selectable icons in the icon bar at the far left. Next to the icon bar, the navigation bar provides a graphical overview and navigation tool for the selected resource. The header list and message view support textual browsing and reading capabilities for the resource. Audio updates add a non-visual cue that alert the user when a change occurs.

Figure 2: A time-lapse series of shots of the Irwin email icon. When new email arrives, the icon changes color, then gradually fades toward the original color. It does not completely return to the original color until the user reads the new message or clicks on icon to acknowledge that it was received. Other icons behave similarly when updates occur, and some change their appearance based on the nature of the contents.

Irwin uses a 16-by-16 pixel animated icon that changes color and appearance based on the state of and changes to the contents of the information resources they represent. In addition, for some types of information the bitmap itself changes based on the state of the resource. The expectation is that changes to the icons will draw attention and convey information about the status of the resource. Figure 2 further explains how the animation occurs.

3.1.1.2 Auditory cues

Auditory cues provide an additional mechanism for alerting users that a change has occurred. They may be the only way the user knows about the change if the display is obscured or if the user is not looking at the screen. Auditory cues have proven useful in both enhancing and replacing visual cues in user interfaces [8, 7].

Based on these previous results, I felt it would be useful to allow users to combine an auditory cue with the visual icon change. I selected sounds that are distinctive but are not distracting and, most importantly, are short. Each is less than two seconds in duration, which I expected would be long enough to be noticed but not too long to become an annoyance. In the initial configuration, I tried to select sounds that were indicative of the resource; for example, a dog bark for mail (dogs always bark at the mail carrier) and a splat for news (similar to a newspaper hitting a front porch). Of course, the user can select any desired sound; Irwin provides a list of twenty.

3.1.1.3 Graphical representations of textual information

Since screen space is at a premium in Irwin, use the space occupied by a scrollbar not only is used for navigating the list but also for providing an overview of the contents of the list. To accomplish this, several graphical methods were developed for encoding useful information about a line or word of text into a small area.

One way to representing large amounts of textual information in a small space is to reduce the size so that text appears as a few graphical pixels. Fisheye views, read wear and edit wear, and the SeeSoft and RunView packages all use this type of encoding [26, 33, 20, 49]. Irwin uses a similar representation that combines this type of information encoding with the power of a scrollbar in a device called a navigation bar. Each line of text in a list is represented by a line of pixels, and the lines are colored and positioned to convey information about the items in a list.

The navigation bar supports both semantic and syntactic encoding of information. Semantic encodings are used when the user wants to highlight specific items such as sender names for email, subject for news, or topics on Web pages. The semantic encoding requires the identification of a word or phrase to encode by analyzing the structure of the sentence using various information retrieval techniques [24]. The semantic encodings require user specification of color mappings or require a legend to explain to the user the meanings for each color. Since a legend takes up a lot of space and space is at a premium, a syntactic encoding was used instead.

The syntactic encodings encode words or phrases from a list graphically, where a 4-by-4 block of pixels represents each character in the word. To differentiate between words of equal length, the blocks are colored that correspond to vowels such that 'a' is red, 'e' is orange, 'i' is yellow, 'o' is green, and 'u' is blue. Since syntactic encodings remain consistent between sessions, users can learn and remember encodings between sessions. Also, unlike semantic encodings, syntactic encodings do not require the user to select item characteristics of interest and thus have a lower startup cost.

Users interact with the navigation bar to control the portion of the information list that is visible in the text view. The black box surrounding a series of messages indicates that they are visible in the textual list. A grey highlighting line indicates the pattern for the currently selected message. Figure 3 shows the navigation bar and textual views displaying information for an email folder.

The time at which the message was stamped is encoded by indenting the message

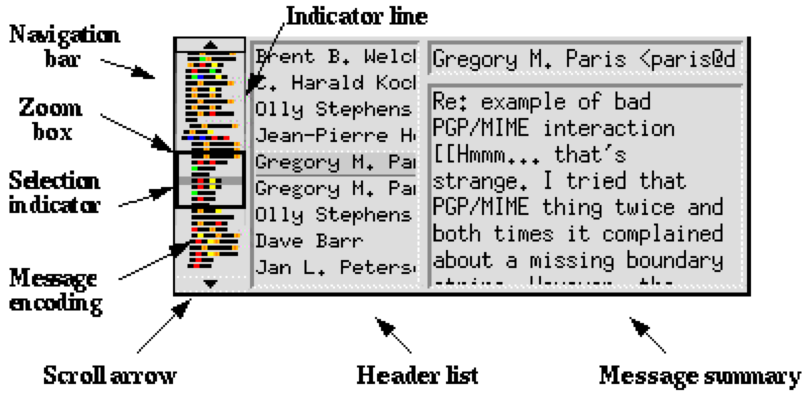

Figure 3: Irwin's display of an email folder. Note that many of the visual structures found in a scrollbar are present in the navigation bar. They function in a similar manner. The message encodings are contained in the trough of the navigation bar and contain a set of coded lines representing each email message. The encodings are indented proportional to the time at which the message was received to group them visually by arrival time. Since not all message representations can fit in the navigation bar at a time, the indicator line reflects the position and percentage of message representations that are visible. The text views to the right provide detailed information about the selected messages.

according to the hour; for example, messages received at 3 PM will be indented three blocks. This is intended to group messages by arrival time, thus facilitating searches and browsing. If users knows the time at which a message arrived, they can identify the range of messages in which it must fall using the navigation bar.

I conducted a study to examine the effectiveness of navigation bars. The study showed that users are able use navigation bars to perform various searching and browsing tasks more effectively than with a traditional scrollbar [50].

3.1.1.4 Textual views

Graphics alone might not show all the information necessary to understand a message — at some point a textual view will likely help. Irwin incorporates two textual views:

a header list and a message view. Irwin constructs lists of headers for each resource, the senders for email and Usenet news, abbreviated headers for Web pages, and days of the week for the weather forecast. If the user selects an item from the list, the message view shows a more informative summary of the original message. The user can click on the message view to jump to a browser to see the full message.

3.1.2 Anticipated Irwin tasks

Irwin was designed with four potential tasks in mind, browsing, searching, monitoring, and awareness.

- **Browsing** - In this task, the user scans through old messages looking for items of interest. I expect that users will notice repeated patterns in the navigation bar that indicate repeated messages with the same topic or author. I hope that certain patterns for desired topics will become familiar to users to the point where they can identify and select these items while browsing.

- **Searching** - In this task, the user searches for a particular message, perhaps related to a recently received message. This task involves matching two or more patterns; for example, a selected pattern with one that occurs earlier in the list. I hope that users will perform searches in Irwin when they are reminded of a previous message by a newly arrived one.

- **Monitoring** - This task involves examining Irwin for specific information, generally when an auditory cue or visual change has occurred. Irwin provides quick access to the most recent changes during monitoring by listing the items in reverse chronological order with the newest item selected and on top of the list. I expect monitoring will be the most frequently performed task since it requires little physical and cognitive effort.

- **Awareness** - In this task, users wish to raise their general knowledge level on certain topics, but they are not necessarily looking for any specific information and they do not intend to interrupt their current activities when they see new information of interest. Awareness can provide peace of mind, and peripheral awareness using Irwin could substitute for more time-consuming browsing with a Web browser.

3.1.3 Case study

In the case study, I wanted to learn how people use technology to increase information awareness. What tasks do users want to do with an awareness tool? Do icons that change appearance provide useful information? Do auditory cues increase awareness in a positive way, or will they become intrusive or annoying over time? Are syntactic encodings usable and learnable? By offering Irwin to users and observing how they use it, I hoped to answer some of these questions.

3.1.3.1 Methodology

Nine people attended a presentation and demonstration explaining how to set up and use Irwin. Seven people tried Irwin at least once, and four of those continued to use it regularly.

I asked those who did not use Irwin why they chose not to. Most simply did not feel that they needed such a tool. One person noted that she didn't receive much email and only rarely browsed the Web: "I'm an example of a fish who doesn't want a bicycle for Christmas - I just don't have much use for it!" One user wanted a less passive visualization tool. He reported the he felt he did not have enough control over when the resources were checked. "At times I want to tell it to go get information about a newsgroup or Web page. I don't want to just wait for the regular time interval checks."

Five months after the presentation I visited with the four frequent users individually for about an hour to discover how they used Irwin, and in particular which features were most and least useful. Many of the questions were open-ended, allowing the users to expand upon issues of interest. I also asked them to discuss other possible uses for awareness and monitoring tools. Their responses are summarized in the next section.

3.1.3.2 Individual reports

The four participants in this study were all members of the Georgia Tech Graphics, Visualization, and Usability Center. This allowed me to assume a certain level of sophistication with email, Usenet news, the Web, and interfaces in general. Alan was a faculty member with a private office and his own Sun workstation, while the other three were graduate students who used X-terminals in cubicles and workstations in a shared laboratory.

The participants' real names have been changed to preserve their anonymity, and none of the participants were required to use Irwin or to answer any questions about it. I hoped these assurances helped to generate a realistic experience.

Participant 1 - Alan Alan used Irwin to monitor his email, the Golf Magazine GolfOnline Web site, ESPN television's ESPNET Web sports news wire, and the local Atlanta weather. He placed Irwin in the lower right corner of the screen with all views generally visible.

Alan used a different auditory cue for each resource. "I chose nice short sounds that are moderately loud – long ones get old." Alan relied primarily on the auditory cues to inform him of changes; he infrequently looked to see if the icon appearances had changed and sometimes even had the icon view partially covered with other windows at times.

Alan received around sixty email messages a day, and he always left his email tool running. When he received new mail, he almost always turned to his email tool rather than to Irwin. Alan mainly used Irwin for passive browsing of Web sites that he normally would not have visited very often. Since Web sites change at irregular intervals, Irwin alerted Alan of changes without requiring him to check the site himself. Alan typically looked only at the currently displayed message when it popped up and rarely clicks on the icons or the navigation bar to view other resources or earlier messages. Sometimes he browsed through the textual list and selected messages to view on the message display. When he saw a particularly interesting message, he used Irwin to pull up the full article in his Netscape Web browser.

Since Alan did not use Irwin to browse old email or Web summaries, he doubted he would use any navigation tool, whether it be a syntactic or semantic encoding or even a regular scrollbar. The only desired feature would be a thread display for email that would highlight threads of related messages, a feature not present in his mail reader. This added value might be useful, but as it was he preferred to pull up his mail reader when he needed to do any searching or browsing.

Alan wanted Irwin to monitor other information resources, in particular Unix commands for checking the status of shared hardware resources. Sometimes when printing a large document, he needed a tool to monitor the print queue and to alert him when the job is done. Alan noted that monitoring the status of processes might be useful to see if they become defunct or exit abruptly. He stated that he would be willing to do a one-time configuration if it were then easy to start and stop these type of monitors. In addition, there were certain Web sites with images that Alan would like to monitor; for example, traffic report maps and weather radar pictures. He reported that he would like for Irwin to display a miniaturized version of these images.

Participant 2 - Bert Bert used Irwin to monitor the local weather and six different email folders. Since Bert filtered his email, he needed a tool like Irwin to help keep track of changes in each folder. Bert configured Irwin so that only the icons are visible unless the cursor was inside the Irwin window. Bert placed his Irwin at the upper left corner of his screen above his clock and system load monitor.

Bert had the same auditory cue (a beep) for all of his email folders. When he heard it, he looked at the Irwin animated icons to see which folder received email. "It's less trouble just to look at Irwin than to remember lots of different sounds." Bert liked the fact that animated icons fade over time because it reminded him that there was email that he had been neglecting. He liked the changing weather icon but was unsure about having other icons change appearance. "Just so it's simple and makes sense."

Bert used Irwin primarily to alert him when new email arrives. He liked the quick access to information that Irwin provided. "It's much quicker than starting email, so I use only Irwin when possible." Bert pulled up his mail reader when he needed to read long messages or to reply to a message, functions that are beyond the scope of Irwin. Often times he knew from the message summary that he could ignore the email until a more convenient time.

Bert generally did not look at the navigation bar, though sometimes frequently repeated patterns would catch his eye. He never tried to do any searching or browsing with Irwin, opting instead to use his email reader and Web browser for such tasks. He reported that he might use the navigation bar more if the information provided were more useful, like perhaps showing the importance of the message in some way. He actually found the weather navigation bar's encoding of the day of week more useful than the syntactic encoding.

Bert reported that he would like to configure Irwin to monitor people's activities, in particular to see if they are available. At the time, he had to run the Unix finger command repeatedly to see when people were at their machines. This type of repetitive task would be well-suited for a tool like Irwin. Bert also wanted his clock and system load monitoring tools integrated into Irwin to save room on his desktop. He noted that some operating systems provide these types of toolbars, but often they are not very configurable.

Participant 3 - Carl Carl used Irwin to monitor his email, a Usenet newsgroup, the ESPNET news wire, and the local weather. He placed Irwin at the top of the screen next to his clock, and he always left all of the Irwin views visible.

The only sound Carl configured in Irwin was a beep for email. He listened to a few of the sounds but decided against using them. "They would drive me up the wall." Carl even complained that some of the longer and louder sounds from other people's Irwins disturbed him.

Instead of sounds, Carl relied on the icons for information about new messages. He particularly liked the weather icon and wished that other icons changed appearance as well. Since he often used an X-terminal with a black-and-white monitor, neither the change in color of the icons nor the syntactic encoding of keywords were of much help. He would have liked to associate icons with different events so that when the event occurred the corresponding icon would be visible.

Carl tended to use Irwin for browsing more than the other users. However, he said he only recognized two patterns in the navigation bar (one being his own encoded login name). Occasionally Carl would notice repeated patterns in some newsgroup representation; generally this meant a knowledgeable person was replying to a number of messages at once, which made for interesting reading.

Carl liked to read the Irwin textual message summaries for news articles because it was quicker than starting a news reader. However, he was frustrated that articles were truncated after a limited number of words. Usually he wanted to see the entire message without having to jump to his news reader. Similarly for Web pages, he did not want the slow-loading graphical view used by most browsers and preferred a quick view of the text only. For Carl, the big advantage of Irwin was the speed at which he could obtain information.

Carl believed that a variety of views in the navigation bar would be more useful. At times he would have preferred a site-of-origin view where the bars are colored by the site at which the message originated, and other times he wanted a message threads view. "It should be easy to configure to see alternate views." In addition, Carl wanted to be able to zoom in and out of the representation, or perhaps even have a fisheye view of the resource. He was concerned that he could not see the entire resource at once.

Participant 4 - Dora Dora used Irwin to monitor her email, USA Today's Washington DC news wire, the local weather, and two Usenet newsgroups. She placed Irwin at the bottom of the screen between her system performance and desktop management tools.

Dora associated a different sound for each category of resources; for example, Irwin barked when new email arrives and beeped for any new news. She would have preferred to combine related resources into a single icon. Then she would have known at a single glance whether a resource of immediate interest had experienced changes. "At busy times I might check my email but not my newsgroups. It would be helpful to only have to check a single icon."

Dora was the only participant who used semantic highlighting. She created three lists of names for her email: one for coworkers, a second for family members, and a third for her PhD committee members. Each list of names had a different highlight color, so when she looked at the navigation bar she could tell who had been sending her email. "When I see a lot of red, that generally means I'm in trouble with somebody." Dora still left the syntactic encoding on as well so she could distinguish between other messages. However, she did not recognize any names, though she sometimes noticed repeated patterns resulting from several messages from the same person.

Since she generally worked in a lab or cubicle without windows, Dora enjoyed the changing icon of the weather display. "It serves as a reminder of what's going on in the real world." She would have liked to see something similar for other resources because it could have provided easily accessible information - a user would only have to look at Irwin to learn about the state of the resources.

3.1.4 Lessons learned

Based on the observations of Irwin users, a number of issues emerged which should help in the future development of information awareness tools.

In general, users were willing to sacrifice a little bit of screen space to increase their awareness of the contents information resources. Even those who did not use Irwin cited reasons other than space limitations. Most users wanted to be able to integrate other monitoring tools into Irwin as well to facilitate smooth transition between the monitoring and awareness tasks performed with with Irwin and the reading and browsing tasks performed with other programs.

Some of the users stated that they would notice repeated patterns in the syntactic encodings when browsing resources. However, none of the users could associate more than a few encodings with the corresponding word or name, and none tried to do any searches using the encodings. Users did not take the time to look at the patterns because they were comfortable using other tools for browsing and searching. Both of these tasks benefit from a large amount of screen space and are not practical in small peripheral tools like Irwin. The main reason people used Irwin for tasks involving a significant amount of reading was to avoid application startup, download, and display delays.

When using Irwin, the primary task performed by the users was to maintain a general awareness of breaking information. Occasionally users took advantage of the auditory cues and icon animation to know when to check a resource. The most significant informational gain from Irwin came from the hands-off textual views.

Unlike with traditional biffs and similar tools, Irwin users can see textual information about new email and news articles, and unlike with traditional readers and browsers, Irwin users can obtain information without mouse or keyboard input. The hands-off, peripheral nature of Irwin provides users with the additional information necessary to make smarter decisions about whether to interrupt their current task to attend to email or other informational tasks.

3.1.5 Thesis focus

The research described in this chapter has examined different communication mechanisms and different usage approaches for peripheral displays. Perhaps the most important advance gained in building and evaluating the Irwin system is the understanding of the possibilities for peripheral displays.

Users seemed most successful at using Irwin for monitoring and awareness, but as soon as browsing, searching, or reading became their focus, other tools that can make full use of the screen seemed more suitable. The tasks differ in that browsing and searching are active tasks that dominate users' attention (and should dominate their screen space), while awareness and monitoring are much more passive tasks that must be accomplished with relatively little effort. When performing monitoring or awareness tasks, users do not want to follow the overview, zoom and filter, details on demand model that Shneiderman touts as the mantra for visual information seeking

[69]. Instead, they want details that are constantly available without any effort in zooming and filtering.

A related concern is to consider the effectiveness of various communication mechanisms in supporting monitoring and awareness. Graphical encoding techniques have proven useful in the display of static information, but its potential in staying aware of dynamic information seems less clear. Audio is useful for drawing attention to status and informational changes, but it can be annoying and intrusive and seems better suited in complementing other techniques. The animated icons provide a lasting reminder of the type and nature of changes to an information resource, yet the amount of information they can provide is somewhat limited. The textual displays naturally provide an easy method for displaying the significant amount of textual information that is generated, but text can take up a large amount of space and the static nature of the textual displays necessitated user interaction in reading the information.

The approach in this thesis is to augment textual displays with animated effects in an effort to raise awareness of a large amount of information in a hands-off manner.

Animation has been used in a variety of situations, including tickering and changing information on television stations, fading sports and news data on Web pages, and animated images in Web and billboard advertisements. Often it is looked upon as more of a distraction than a benefit, but the potential upside makes it worth considering.

The use of animation has several potential advantages:

- Animation provides an extra dimension in which to communicate information. Thus, more information can be shown in a given amount of space. This is particularly crucial for situations where space is at a premium.

- Animation can attract attention. In a high- or medium-priority situation, it may be desirable to draw the user's attention to the new information. By animating the information, the user will be drawn to the display and will obtain the information in a timely manner.

- Animation can show change in state. As a simple example, percent-done indicators comfort the user by providing a clue as to the progress of a download or save. In some of the situations, animation could be used to show subtle changes in information over time, such as the progress in the Cleveland Browns stadium, by animating a series of information snapshots.

- Animation can show physical actions that are difficult to represent statically. Animated videos often show an action that is not adequately captured in a single frame photo. A study showed that animated icons in a drawing program (for example, an eraser erasing and a paint brush painting) communicate the intended meaning of the icon better than can a static icon [4].

Several potential disadvantages must be considered as well.

- Animation can distract the user from other tasks. This is fine as long as the animated information is more important to the user than the current task, but often this is not the case. Care must be taken to use animation sparingly, and to use subtle animation in non-high priority situations.

- Animation can consume valuable computer resources. Especially on shared machines or when a user is performing compute-intensive tasks, an animation can consume valuable cycles that may be better suited for other tasks. This is particularly a concern with animations that continue repeatedly for an extended period of time.

- Users can experience a lack of control with animation. An important rule of interface design relates to the users' sense of control – it is important to make sure that users feel like they are in control of the information that appears on the screen. Animation seemingly goes against that rule by changing the display in a manner determined by the programmer, not the user. In addition, animation can make it difficult to process information, especially text.

Gradual and repetitive animation shows potential in creating useful and usable information awareness applications. By animating large amounts of information in a small space, the remaining space can be used for other applications. Changes to the information can be integrated gradually into the display in the next iteration, minimizing the disturbance to the user. Constantly cycling through the entire information space lessens the number of physical interactions required to obtain the information – rather than having to press a series of buttons or keys to get information, the user need only wait for it to cycle through.

The next chapter examines the effectiveness of various types of animated textual effects in maintaining awareness.

CHAPTER 4
Evaluating Animation as an Awareness Technique

The goal of this chapter is to demonstrate the viability of animated peripheral displays as a technique for helping users maintain awareness while completing other tasks. The philosophy presented in this thesis is that a constant, repetitive animation can help maintain awareness of dynamic information by requiring little space and by integrating changes smoothly in the next iteration. It is expected that users would be willing to give up space on their computer desktop for a display that cycles between items of interest, and as long as the animation is gradual, the distraction should be minimal. However, animation has been discounted in many situations as too intrusive because the constant motion on the screen may be distracting. Furthermore, it is unclear that the space tradeoff would prove to be worthwhile. However, computer users have long sacrificed desktop space for constantly changing displays like clocks, email biffs, and load monitors. Perhaps they would also be willing to sacrifice space for heightened information awareness.

In considering the use of animation in peripheral information displays, it is desirable to explore a range of animation styles to understand whether they differ in their ability to communicate and in their potential for distraction. One common method for animating information is with a motion-based effect whereby the old information is moved off the display area while the new information is moved on. In performing the evaluations described in this and subsequent chapters, motion-based effects were represented by a *tickering* effect whereby old information moves off the screen to the left while new information moves on the screen from the right.

A second common method for animating information is with an in-place effect whereby the old information disappears and is replaced by the new information. In this category, a *fading* effect was selected whereby the old information fades into the background while the new information fades into the foreground. The expectation in choosing the tickering and fading effects is that other motion-based effects such as vertical rolling will be similar to the tickering effect and other in-place animated effects such as stippling and swiping will produce similar results to those by the fading effect. This is supported by the Maglio experiments described in Chapter 2 that demonstrated that there was no performance difference between participants using vertical and horizontal motion-based effects [45].

Another important factor to explore is the scope and focus of the awareness problem. The preliminary work described in Chapter 3 revealed that a powerful and complex system like Irwin is used in a variety of ways, many extending beyond the scope of awareness. Tasks such as searching, browsing, and reading seem better suited for larger, more attention-demanding interfaces. In evaluating peripheral animated displays, I want to focus on their use in the context of a supplemental activity, with the primary focus on other tasks. The expectation is that people will use the displays only to keep track of the state of information and will use other more suitable tools for searching, browsing, and reading. As such, only the most basic information is provided in the interfaces. Note that this still leaves room for a wide range of awareness-related tasks ranging from using the displays to gain knowledge over time to looking for specific information that is acted upon when it appears.

4.1 Pilot study

I began this research by conducting a pilot study. It provided me with an opportunity early in my research to present peripheral displays to potential users and monitor their usage and collect their opinions. Results from the pilot study helped guide the future studies described in Chapters 5 and 7 and the development of the toolkit described in Chapter 6.

The pilot study examined the use of animated devices in maintaining awareness of sports scores during the opening days of the NCAA Basketball Tournament. This section examines the role of animation in interfaces, focusing on lessons learned in integrating animation into peripheral displays and in observing how users made use of the animation.

To test the utility of animation as an awareness device, I created tkscore, an application that monitored and communicated scores from the NCAA Basketball Tournament using various peripheral displays. I then deployed the application and surveyed users on their likes, dislikes, and future desires for such peripheral information displays.

The NCAA Basketball Tournament is a single-elimination tournament consisting of 64 teams from across the United States. It generates much interest and is recognized as one of the most exciting sporting events in the nation. The large number of teams in the tournament necessitates that some of the early tournament games be played on weekdays. As a result, numerous people who are working have a desire to stay abreast of the scores of ongoing games, an ideal situation for an awareness tool.

The tkscore application shows the scores of basketball games using either a ticker, fade, or list display. The tkscore tool is started with a single command, which brings up a control panel (see Figure 4) listing all current games and the three available display types. The user can select and deselect the games to be monitored by clicking on the checkbox next to them. The user can toggle between the display types by clicking the radio buttons. The control panel can be dismissed by clicking the OK button and returned by pressing button 3 in one of the display windows.

The list display lists the selected scores and updates them as soon as the scores change without smooth animation (see Figure 5). It shows all of the scores at once but can occupy a large portion of the screen. The fade display uses a fading effect to alternate between scores (see Figure 6). The ticker uses a tickering effect to scroll scores across the screen (see Figure 7).

New scores appear immediately in the list display and on the next iteration for the ticker and fade displays. Score updates are obtained every minute from the USA Today Scores Web Page, `http://www.usatoday.com/sports/scores.htm`.

For the study, all games for that day initially were selected and the participants had to choose a display type from a randomly ordered list of the three choices. The participants were encouraged to use several displays for an extended period of time. The informational and display selections of the users were monitored.

After the study, the participants were asked a series of questions relating to their desires for information and their display preferences. A complete list of the questions and a tabulated list of responses is given in Appendix A (including breakdowns of responses based on animation preferences). The remainder of this section discusses

Figure 4: The main tkscore screen used to select display type and games. The display types are presented in random order upon startup to help avoid selection bias. If the fade or ticker display is selected, a speed selection bar appears. Initially all games are selected - a user can toggle the selection by clicking on the box next to the game.

Figure 5: The tkscore list display. All selected scores are visible. When a score changes, the list display is immediately updated.

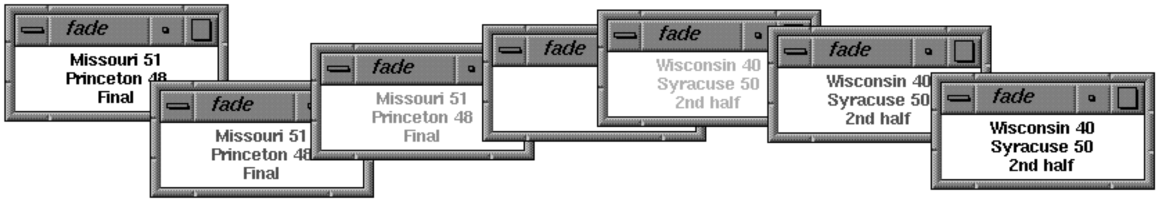

Figure 6: Seven overlapped time-lapse snapshots in the operation of the tkscore fade display. The leftmost, bottom frame shows an initial block of text representing one basketball game. The next three frames show how the text fades away into the background, and the final four frames show how the new text for the next game appears in the same place.

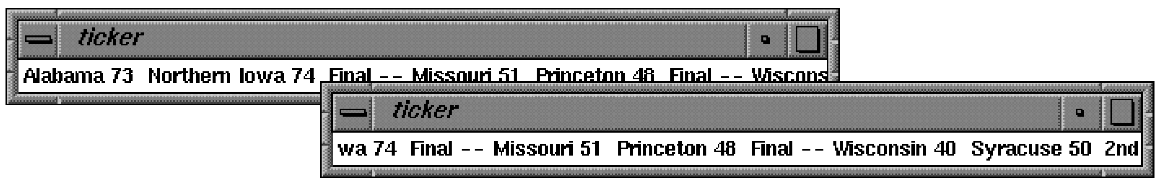

Figure 7: Two time-lapse snapshots of the tkscore ticker display. The scores ticker across the screen, and when a score is updated it appears in the next iteration. The foreground ticker shows the state a few seconds after the background ticker.

some of the key responses.

Twenty-five faculty and students used tkscore during the first four days of the tournament, and twenty-one of those answered the set of questions. Most used tkscore on several occasions, and most checked scores several times a day during the tournament. The usage monitoring logs (obtained for only eleven of the participants because of technical difficulties) showed that most of the eleven participants tried all of the animations, though only five users changed the animation speeds, and the speeds they settled on varied significantly. The cumulative time that a given participant used tkscore ranged from a few minutes to several hours.

Each display type had its proponents: seven people chose the fade display as their favorite, six chose list, and five chose ticker. The other three did not pick a favorite.

Information demand	Overall	Top preference Fade	List	Ticker	No pref
Continual	8	3	3	1	1
Occasional	4	0	1	2	1
Casual	5	3	2	0	0
No real interest	4	1	0	2	1

Table 1: Number of responses to the question "How would you describe your interest in the NCAA Tournament?" from the 21 participants who returned the survey. The leftmost column of numbers shows the overall count for the participants while the other four columns show a breakdown based on the top preference from among the types of displays.

Response	Overall	Top preference Fade	List	Ticker	No pref
More than needed	3	1	0	2	2
Right amount	10	5	3	2	0
Less than needed	7	1	3	1	2
No response	1	0	0	0	1

Table 2: Responses to the question "How did tkscore meet your needs for tournament information?"

Seventeen of the participants had at least a casual interest in the tournament, with eight expressing a desire to know about scores, game status, and more continually throughout the day. A breakdown of the participants' interest level is given in Table 1. Their evaluation of how well tkscore met their informational needs is shown in Table 2.

The reasons for selecting a display varied. The most common reason (given by 15 participants) was the ease with which information could be accessed and read. Ironically, this response was given equally by those who listed fade, list, and ticker as their favorite display. Table 3 shows a breakdown of the contributing factors to

Response	**Overall** (max 21)	**Top preference** **Fade** (max 7)	**List** (max 6)	**Ticker** (max 5)	**No pref** (max 3)
Ease with which information can be accessed and read	15	6	4	4	1
Amount of information visible at any given time	12	2	5	4	1
Level of distraction	9	5	1	2	1
Control over visible information	8	3	2	3	0
Control over rate of change	4	2	0	1	1
Other	4	3	0	1	0

Table 3: Responses to the question "What factors contributed to your choice of display?" Since each participant could select multiple responses, the overall responses total more than the number of participants.

display selection.

When displaying information in a small space, certain choices must be made in constructing the display. The participants were asked to select the other features of the information that they would like to be displayed. Note that auditory cues, often looked upon as distracting, were the most desired mechanism for the users. Generally, the desire was to have auditory cues tied to important changes to the information, perhaps reflecting the desire to know better when and how changes occur. The second most desired mechanism was easier control over speed of updates, and the third was for more potential for user interaction. Both are related to the desire of users to feel in control of the interface, one of the golden rules of interface

Response	Overall (max 21)	Top preference Fade (max 7)	List (max 6)	Ticker (max 5)	No pref (max 3)
Auditory cues tied to important changes to information	8	1	3	3	1
Easier to control speed of display updates	6	1	2	2	1
More potential for user interaction	4	0	2	1	1
Pictures and other graphics	3	1	2	0	0
Other	6	3	2	1	0

Table 4: Responses to the question "What additional display mechanisms do you desire in a program like tkscore?" ordered based on total number of responses. Each participant could select multiple responses.

development. Improvements to the interface that deal with both of these issues are discussed in Section 6.4.1. A complete list of responses is given in Table 4.

Overall, responses to the application were positive. Users expressed desires for similar displays for weather information, news headlines, stock quotes, and activities of friends and colleagues. See Table 5 for a summary of the other informational desires.

4.1.1 Guidelines

In developing the tkscore application and in collecting and analyzing the results of this study, several objectives emerged that should be met when designing peripheral displays for maintaining awareness of dynamic information:

Response	Overall (max 21)	Top preference Fade (max 7)	List (max 6)	Ticker (max 5)	No pref (max 3)
Weather information	12	5	4	2	1
News headlines	11	3	4	3	1
Stock quotes	9	3	3	2	1
College announcements	8	3	2	3	0
Friend/colleague activity	8	4	3	1	0
Email/newsgroup summaries	6	3	2	0	1
MUD/MOO/Chat information	0	0	0	0	0
Other	5	2	1	2	1
Total	57	22	19	13	5
Average per person	2.71	3.14	3.17	2.6	1.67

Table 5: Responses to the question "With what other information sources would you use tkscore-like displays?" ordered based on total number of responses. Each participant could select multiple responses.

- **Communicate state of and changes to information.** As with the communication of any information, communication of dynamic resources requires that the current state of the information be communicated to the user. However, dynamic situations often have the added requirement that users want to know when and in what ways the information has changed.

- **Minimize the space required by the peripheral display.** To put it another way, a peripheral display should maximize the space that can be used for other tasks. Since all the tasks compete for the same computer desktop space, the awareness task, which is of secondary importance, should occupy a minimal amount of space.

- **Minimize required user interactions.** Since awareness is not the sole task of most users, it is undesirable to create highly interactive displays that require user manipulations to reveal needed information. Ideally, a peripheral display should communicate enough information to maintain a sense of awareness with little or no action from the user and with minimal interruption of other tasks.

- **Maintain user's sense of control.** A primary tenet of user interface design is to maintain the user's sense of control [69]. Inability to obtain desired information and unexpected changes in the appearance of the interface can create anxiety and dissatisfaction in the user. This seems to conflict with the problem at hand: since the resources considered in this thesis change frequently, it is necessary to update the display to reflect these changes.

These guidelines emerged from personal experiences and from user comments before and during the development of the tkscore interface. To simplify the design and implementation of applications that conform to these guidelines, support for them was included in the development of my animation toolkit. In the toolkit, highlighting techniques help communicate the state of and changes to information. Support for cyclic animation helps minimize space and required user interactions. Mouse bindings help users maintain a sense of control over the information flow. The toolkit is described in Chapter 6.

4.1.2 Discussion

In the introduction to this chapter, several issues were raised concerning the nature of the awareness task, the role of animation in peripheral displays, and the relationship between the awareness needs of a user and the type of peripheral display that should be employed. This section addresses these issues.

4.1.2.1 Defining peripheral awareness

Unlike the Irwin peripheral device, tkscore did not have support for browsing, searching, and in-depth reading of information. As such, people had to be content with awareness-related tasks. They could obtain information on scores, but for deeper analysis they had to turn to full-screen browsers (a more suitable interface anyway).

There was still a wide range of awareness-related tasks that were undertaken by the study participants. The high importance that many people attach to the tournament meant that many of the participants wanted a great deal of information close at hand. Many of them wanted even more information than was provided, such as the time remaining and high scorers. As soon as a game was over, often they would temporarily stop their primary task to pursue more information such as the game story and box score. This awareness task borders on information monitoring, since it occupies a high priority for the users and can cause them to interrupt their current task.

Some participants were content with the occasional glimpse at a score and did not need to pursue the information any further. These participants wanted to learn over time but were not willing to interrupt their current task to pursue the information. Many of them did not have a high degree of interest in the tournament but did want to know about results at the end of the day.

Of course, the participants cannot be divided into two distinct groups. Most would passively allow the display to run, but at certain times between primary tasks they would interact with the display, perhaps pursuing more information on a surprising result. I expect that this is how most people would interact with a peripheral animated display — at most times they would simply let it run, but occasionally an item of interest would spur them to seek out more information or perform some task related to the item. For this reason, the empirical evaluations described in Chapter 5 analyze performance on intermingled monitoring-style and learning-style awareness tasks.

4.1.2.2 Understanding the role of animation in awareness

Before undertaking this study, I had several questions about the role of animation in the awareness process. Would people be willing to sacrifice at least a small amount of space to potentially heighten their information awareness? Can animated displays in fact raise awareness to a suitable level? Or would the animations prove to be too distracting? The pilot study was able to shed some light on these questions.

The first issue is whether people would be willing to sacrifice screen space for a potential gain in awareness. It is obvious from the extended use of tkscore that people were indeed willing to sacrifice a bit of screen space. Some people were even willing to have the entire list at once, though this was one of the reasons one participant gave for not using the list display. In the comments section of the questionnaire, no one mentioned space concerns as a reason for not using the tkscore application.

Note that many users preferred the list display, which required significantly more space and did not animate changes. However, the sacrifice in screen space was still far less than for other options (such as checking the scores in a browser or starting many of the large Java monitoring applets), and the level of distraction was not listed as a major contributing factor in choosing a display for those who favored the list display (see Table 3). Those who were most concerned about being distracted by the display chose the small fade display that included perhaps the most gradual and least intrusive animated effect.

The second issue is whether these types of applications actually raise a user's awareness of the dynamic information. Over half of the tkscore users conveyed satisfaction with the information provided, and those that were not satisfied typically wanted more information to be displayed (the time remaining in the game or the players with the most points). While it is unclear whether users actually learned anything or whether they were able to accomplish other tasks, the users' satisfaction with their awareness levels is one important factor. Several of the most avid fans seemed to need more control over the rate at which new information was obtained: they repeatedly hit the reload button for a Web page, making the awareness of scores a primary rather than secondary activity.

Another concern is that a constantly changing display will be too distracting to the user. I attempted to evaluate this by asking users if the level of distraction was a factor in selecting a display. Five of the six users who preferred the list display also stated that level of distraction was a significant factor in selecting a display, which seems to indicate that some users prefer occasional sudden updates to the display over continuous animation. However, five of the seven fade proponents and two of the five ticker proponents also listed distraction as a significant factor, so there may be other factors in effect.

Maintaining user control over information flow is well-established as an important factor in designing visualizations and interfaces. None of the users listed it as a major motivating factor in selecting an interface, though some listed it as desirable in future versions of the tool. The main concern was the ease with which more information could be obtained. An action that allows the user to jump to a Web browser showing details of the game of interest may best solve this problem. It is important to recognize the bridge between maintaining awareness and obtaining information, and to allow separate modalities to accomplish each.

4.1.2.3 Matching animation types to awareness tasks

One goal of this research is to understand better the communication strengths and weaknesses of the types of animation introduced in this thesis. Do the animations differ in their ability to communicate information, and if so, what types of animations are best suited for different situations? One indication of the relative qualities of the animations can come from a comparison of the informational preferences of the participants who favored each device. The relatively even division of top preferences (seven for fade, six for list, and five for ticker) suggests that no single animation type is best suited for every person and every situation.

Consider the question that asked for additional desired display mechanisms. One of the options was audio, generally considered to be highly intrusive. Interestingly, the data reveal that only one of the seven fade proponents wanted auditory cues, compared to three of five ticker proponents and three of six list proponents (see Table 4). This suggests that perhaps the fade proponents are less willing to be disturbed by audio, suggesting that perhaps they chose the fade animation also because it was less intrusive or less cognitively demanding.

The list display was the only one that did not use smooth animation. It also takes up the most screen real estate. For these reasons, it is more like a standard application designed to command the attention of the user and not like an awareness device intended only to provide secondary information. I suspected that the participants who favored this display were extremely interested in the scores and were willing to sacrifice physically and mentally to stay up to date. The data bear this out. Table 2 shows that half of the list proponents found that the list, while their favorite, provided less than a desired amount of information. Compare this to fade (one of seven) and ticker proponents (one of five). The comments of the list proponents seem to further support the hypothesis. One noted that "I wanted info besides just the score, like pertinent stats on key players". Another said "It was much easier to use CBS Sportsline", a Web page that provides a large display showing significant information.

4.2 Followup work

One area of interest going into the pilot study was the scope and focus of the awareness problem. The pilot study described in this chapter revealed that people will use peripheral displays for a wide range of awareness-related activities. Some want to use peripheral displays to closely monitor changing information, often waiting for certain occurrences that will impel them to do some other activity related to the information. Others choose to learn about information in a hands-off manner. At the end of a period of time using a peripheral display, their desire is to know more about the information that was in the display without having had to seek it out.

A concern from earlier work was that the difference in information type was the primary reason for differences in animation preferences. Since the type of information in the pilot study was held constant yet the users' animation preferences still differed, the difference must not be due solely to the information but to other factors such as differences in the communication properties of the various characteristics of animation, or differences in individuals. This information can be used to better match animation characteristics to awareness tasks and to user populations.

One potential reason for the differences in preferences could be differences in the communication properties of various types and characteristics of peripheral displays. Perhaps one type of animation, say a fading effect, is fundamentally better than another for certain types of tasks. Perhaps there is an ideal size for a peripheral display. The empirical evaluations described in Chapter 5 explore these and other possibilities.

Also important in the understanding of the role of animation in the awareness process is its integration into applications in ways that are most helpful to different people with a variety of needs. In so doing, it is important to understand how certain characteristics of individuals can shape their willingness to use and benefit from animated displays. As a hypothetical example, a person who spends long hours at the computer might prefer the ticker. The observational study described in Chapter 7 was conducted to examine this possibility. Based on the knowledge gained from the pilot study, I expanded the interface to include more informational options and more control over the display options. The new tkwatch interface can display news headlines, stock quotes, weather information, and sports scores. The user can control the color, font, and size of the display as well as the presence of history-based shadowing and automatic highlighting described earlier. The study examines the correspondence between display choices and informational desires for potential users and categorizes users based on observed and reported behaviors.

CHAPTER 5
Empirical Evaluation of Peripheral Animation

While one might suspect that one display type, size, and speed would be intrinsically better and a clear favorite for a given task like basketball score monitoring, the pilot study revealed that users of peripheral animated displays have relatively equally distributed preferences for animation type, the size of the display, and the speed of the animation. This suggests that the preferences are perhaps related to the way in which people want to acquire information rather than simply to the nature of the information itself. This chapter examines whether characteristics of an animated display impact information acquisition during the awareness process. For instance, one might hypothesize that a fading animation might be better suited for watching for a specific change, a slow display might be less distracting to a user who is more interested in other non-peripheral tasks, and a large display might aid the user in processing and remembering displayed information over time. If results such as these are shown to hold, there would be a better mapping between the goals of a user and the characteristics that should be incorporated into a peripheral display.

5.1 Overview of the experiments

To examine whether animated displays impact information acquisition during the awareness process, two empirical evaluations were conducted. Participants were asked to complete a central activity while simultaneously watching a peripheral display showing constantly changing news, weather, stock, and sports information. As the central activity, participants performed a series of *browsing tasks* where they searched a hypertext environment for the answers to questions. To motivate participants to watch the peripheral displays, participants were asked simultaneously to perform *monitoring activities* that asked them to press a button when they noticed specified information in an animated display at the top of the screen. At the end of each round of monitoring activities and browsing tasks, participants would answer a series of *awareness questions* to test the memorability of the information that was displayed. The layout for the experiments can be seen in Figure 8. The motivations for these experimental choices follow.

5.1.1 Browsing tasks

A central activity is necessary because in realistic settings, individuals would not spend their time staring at an animated display but would focus their attention elsewhere. Most prior studies examined reactions to standalone displays [19, 28, 37]. For the one that did not, the primary task was an editing task [45]. From the pilot study and the observational study, it was noted that many users do not prefer to use animated displays at all times, but rather they only use them during certain low-cognition tasks like browsing the Web or looking at email. As such, it seems appropriate to choose a similar task as the primary task for this experiment. As a central task in both of the experiments described here, the participants performed browsing tasks that required them to search in a hypertext environment for specific information.

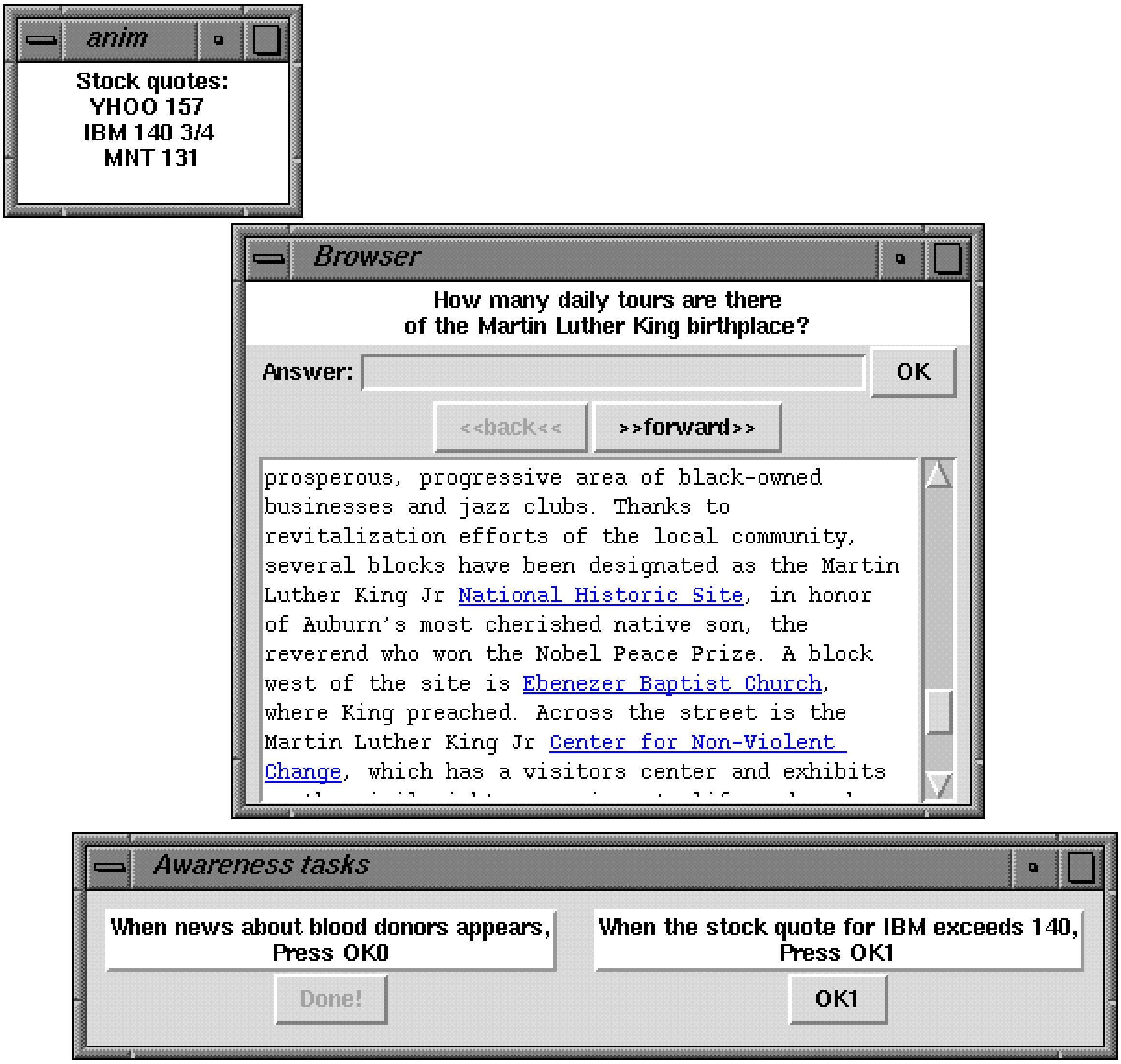

Figure 8: Layout of the experimental environment experienced by participants. At the center is the browser used by the participants in the experiment. At the top of the screen is a peripheral display that cyclically showed the state of several types of information. At the bottom is the area used for monitoring activities. After each round, the screen cleared except for a question area where the awareness questions are presented.

In performing the browsing tasks, participants used a simple browser and hypertext pages. The browser consisted of an information area containing shortened versions of pages taken from World Wide Web sites. The text-only information area contained highlighted, underlined links that pull up other pages when clicked with the mouse. The participants navigated the information space by clicking on highlighted underlined links and with forward and back buttons. These buttons work just like corresponding buttons on browsers like Netscape Navigator and Microsoft Internet Explorer. The browsing tasks were non-trivial: the participants had to read and navigate through a hypertext space to find certain information in the pages, enter it into a box connected with the browser, and press a button to continue.

To minimize the typing required, all solutions were numerical (for example, "In what year was Mount Rushmore carved?") If an incorrect answer was entered, the interface beeped and the participant had to continue searching until the correct answer was entered. When the correct answer was entered, the participant could proceed to the next browsing task.

5.1.2 Monitoring activities

While performing the browsing tasks, the participants were asked to monitor the contents of a peripheral display. The displays included an in-place sudden rapid serial visual presentation (RSVP) display referred to here as a *blast* display, an in-place gradual *fade* display, and a gradual motion-based *ticker* display.

The participants used information in the peripheral display to complete a set of monitoring activities and to answer a series of awareness questions. By including both short-term monitoring-style awareness activities (called monitoring activities) and longer-term knowledge-gain questions (awareness questions), these experiments simulated a wide range of informational situations that could occur on a daily basis. For example, people need to monitor traffic incidents, weather disturbances, and important email that may affect their current behavior at least in the short term, yet they also may want to keep track of temperature patterns, sports scores, and news headlines that may be of general interest but not to the point that it would change their current behavior.

The peripheral display cyclically showed instances of different types of information, such as a ball score, a stock quote, and a weather report. Each instance was updated frequently as it would be in real life. In the monitoring activities, participants were asked to press a button when the information in the peripheral display matches some criteria (for example, "When the temperature drops below 35, press OK1.") The information that was selected for display was interesting but rarely vital, and the informational occurrences that were selected were selected because they might spur a user to perform some activity, such as bringing in a plant that is outdoors or selling a stock that is performing poorly.

Each round included two such accompanying monitoring activities. If the button was pressed at the correct time, it was greyed out to alert the participant that the task had been completed successfully. If the button was pressed too soon, the interface beeped and the button remained active.

5.1.3 Awareness questions

At the end of each round of monitoring activities, the participants were given awareness questions that asked them to recall information that was shown in the peripheral display. The questions were multiple-answer multiple-choice questions that addressed both content and temporal issues. Each question had four possible answers, and there was always at least one correct answer. Generally there were either one or two correct answers, but sometimes there were as many as three.

The first question always presented four types of information and asked the participant to choose the ones that had been displayed. Later questions asked about details of the information, such as which news stories appeared, which stock quotes constantly increased, or which sports team scored the most points.

Initially all of the answers were unselected. The participants clicked on the answers to select them, then clicked an OK button to move on to the next question or complete the question session.

5.1.4 Data collection and evaluation

To compare performance among groups, the times for all browsing tasks and monitoring activities and the answers to the post-round awareness questions were recorded. The results were analyzed to determine whether differences in certain measures occurred for different groups of participants (participants using different peripheral displays in the first experiment, and participants using different sizes and speeds in the second).

In instructing the participants, it was emphasized that fast performance on both browsing tasks and monitoring activities was important. The participants were informed that questions about the information in the peripheral display would be asked, and they were given sample questions to prepare.

The *browsing time* is the time from which the browsing task and browser information appeared on the screen to the time when the participant typed in the correct answer and pressed the OK button. Since the order in which browsing tasks were presented was held constant for all participants in each experiment, the average browsing times for a participant in a group could be compared.

For the monitoring activities, the *monitoring time* is the time from when the information was first entered into the cyclic display until the information was acknowledged as seen by pressing a button. Again, the order in which monitoring activities were presented was held constant for all participants for each experiment. Since participants did not have to complete all activities, the number of activities that were completed in each round were compared between groups.

For the awareness questions, the participants' responses to each of the four answers were collected. I considered each question as being worth four points: correctly or incorrectly assessing each possible response to a question.

A number of different methods can be used to determine a participant's ability to recall information. The most obvious measure for determining responsiveness is to compare the percent of correct responses for the awareness questions in different situations. The percent of correct responses is referred to as the *correctness rate*.

The correctness rate measure potentially can misrepresent a participant's awareness of information. Note that a participant who did not remember seeing anything in the peripheral display and left all responses unchecked would have a correctness rate as high or higher than a participant who remembered seeing several items but was mistaken about what was seen and checked the wrong box. Consider as an example the situation where a question asks a participant to select items that appeared in the peripheral display, where three of the selections were present and one was not. A participant who recalled seeing three items in the display but only was correct about two of them would have a correctness rate of 50 percent, identical to the correctness rate for a participant who only recalled seeing one of the items that was present.

An alternate measure for determining responsiveness is the *hit rate*, a term from signal detection theory defined as the ratio of correct stimuli responses to the total number of stimuli. In the situation described above, the hit rate for the participant who recalled seeing three items but was only correct about two of them would be 67 percent (two of three stimuli were correctly identified) while the participant who correctly recalled seeing only one item would be 33 percent (one of three stimuli were correctly identified). Since a typical goal of using a peripheral display is to proactively recall seeing information, it may be better for a participant to be mistaken about seeing information that was not displayed than to be mistaken about not seeing information that in fact was displayed.

The hit rate is hard to interpret unless accompanied by the *false alarm rate*, the ratio of incorrect stimuli responses to the total number of times when the stimuli was not present. In the situation described above, the false alarm rate for the participant who recalled seeing three items when only two were present would be 100 percent (the one item that was not present was incorrectly identified as being present). As one might expect, a higher hit rate is often accompanied by a higher false alarm rate as well. This work will consider whether this is the case for awareness situations.

The hit rate and false alarm rate are important in situations such as airport security detectors, where more hits means that a bomb or other illegal device is less likely to pass through undetected. One can certainly see parallels in information awareness. Perhaps a person wants to remember that a story occurred, or that a tornado watch is under way, or that a traffic bulletin appeared. The hit rate would reflect the awareness potential of an animated display.

In analyzing the results, analyses of variance (ANOVAs) were performed to check for statistical significance among different conditions of the experiments. ANOVA is a test performed when there are more than two conditions in the experiment. A description of ANOVA and its advantages can be found in most statistics textbooks, see for example [29].

When reporting ANOVA statistics, three values traditionally are provided:

- $F(x, y)$ represents the ratio of variance among groups to variance within groups (that is, individual subject differences). The x and y values reflect the number of cases and the number of participants, respectively.
- MSE is the mean squared error.
- p is the probability that the difference occurred by chance. Traditionally if this is less than 0.05, the results are said to be statistically significant.

If the ANOVA revealed a significant difference, pairwise t-tests were performed to determine which conditions differed. For t-tests, the p value is given. Again, a more detailed discussion of t-tests can be found in most statistics textbooks (see [29] for example).

5.1.5 Chapter overview

The remainder of this chapter describes the two empirical evaluations that were performed to explore the relative advantages of different animation characteristics. The first experiment, described in Section 5.2, compares relative performance when using fading, tickering, and blasting displays as well as when no animation at all was present. The second experiment (Section 5.3) explores the impact of changes in the size and speed of the animated display. Section 5.4 examines participant reactions to animated displays collected after completing the experiment, and Section 5.5 provides a general discussion and some recommendations gleaned from the two experiments.

5.2 Experiment 1

Will peripheral animation indeed distract a user from other tasks? Or can it help with this emerging awareness task? Are different animations better suited for different information types? To address these questions, an experiment was conducted to examine the awareness capabilities of three types of peripheral displays: a motion-based effect that moves or *tickers* information across the display area, an in-place *fade* effect that gradually changes information at a fixed location, and an in-place sudden RSVP or *blast* effect that rapidly changes the visible information without the benefit of smooth animation.

5.2.1 Method

This experiment focused on three factors: the possibility for degradation in performance on a browsing task when an animated display was present, the speed in identifying and reacting to changes in peripheral displays, and the ability to remember information that appeared in a peripheral display.

Seventy undergraduate students participated in this experiment for class credit.

The experiment was conducted on Sun Sparcstation 2 workstations, each connected to a 15-inch monitor with an optical mouse. Participants were run in small groups, one participant per computer. The experiment was explained to each group verbally and again on the computer with examples.

The participants performed six rounds of browsing tasks, monitoring activities, and awareness questions. In each round, participants completed four browsing tasks using a simplified browsing environment. During each round, participants performed two monitoring activities using either a fade, ticker, or a blast animation. The speed with which the information was displayed corresponded to the mean speeds selected by the participants in the pilot study described previously. While this resulted in different rates of information display for the animations, I felt it was a more realistic and ecologically valid measure of how people would use them. The ticker continually shifted one pixel every 50 milliseconds, while the fade and blast updated their entire contents every 2000 milliseconds. The fade required 500 milliseconds to fade between items, while the blast updated instantaneously.

For the first three rounds, the peripherally displayed information was a mixture of sports scores, stock quotes, weather information, news headlines, and email sender names. The latter three rounds each focused on a single type of information: stock quotes, news headlines, and sports scores, respectively. At the end of each round, the participants were asked awareness questions about the information that appeared in the animated display. The first question asked which types of information appeared in the display. For each correctly-identified instance of information appearing, two questions about the information were asked up to a total of five questions. See Table 6 for sample questions and responses.

	Question	A	B	C	D
1	What types of information appeared in the animated device?	Weather data	Local news	Stock quotes	Sports scores
2	Which weather statistics appeared in the display?	Temperature	Visibility	Pressure	Humidity
3	Which weather statistic appeared in the display but did not change?	Temperature	Visibility	Pressure	Humidity
4	Which stock quotes appeared in the display?	AXP	IBM	AOL	DIS
5	Which stock quote constantly increased?	AXP	IBM	AOL	DIS

Table 6: An example of the awareness questions and responses

The independent variable was the type of animation used. This was a within-subjects experiment: all participants experienced all animations, with orders based on a latin square design (browse - fade - ticker (n = 17), fade - ticker - browse (n = 17), or ticker - browse - fade (n = 21)). A different animation was used for each of the first three rounds with the order repeated on the last three. In addition, one group (n = 15) did not have any animations present at any time and as such performed only the browsing tasks.

The dependent variables were the completion times for the browsing tasks and monitoring activities. The completion times for the browsing tasks were measured from the start of each round (for the first task) or from the completion of the previous task.

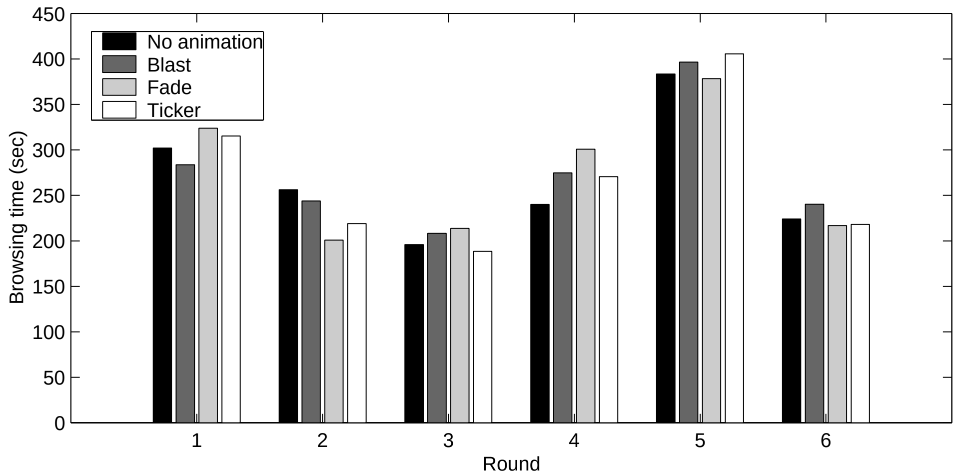

Figure 9: Average completion times for browsing tasks for each round based on the type of animation that was present. Participants performed about the same on the browsing tasks regardless of the type or even the presence of animation. By showing the rounds individually, one can see that there is not even a trend to suggest that participants performed better in certain cases.

5.2.2 Results

This section examines the participants' performance on the browsing tasks, monitoring activities, and awareness questions. For the performance times on browsing tasks, there was not a significant impact due to the presence of a peripheral animated display ($F(3, 58) = 0.60$, $MSE = 46277.71$, $p = .62$). Furthermore, the type of animation did not affect the browsing times ($F(2, 46) = 0.62$, $MSE = 25411.63$, $p = .54$), (see Figure 9).

For the monitoring activities, the number of tasks completed does not depend on the type of animation. On average, participants completed 11.3 of the 12 monitoring activities, including 3.83 of 4 for blast, 3.79 of 4 for fade, and 3.71 of 4 for ticker, not a significant difference ($F(2, 96) = 0.77$, $MSE = 0.25$, $p = .46$). That is, it does not appear that a participant is more likely to identify (or miss) a piece of information

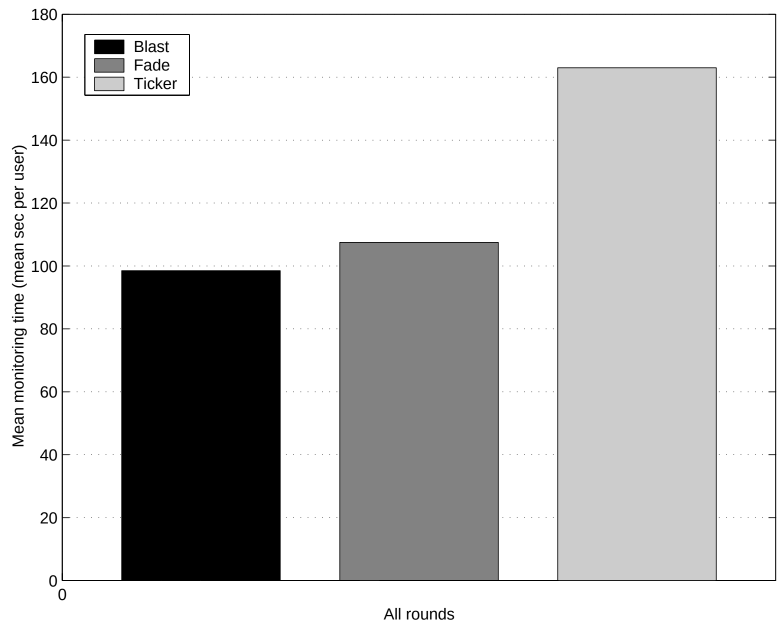

Figure 10: Average completion times for each monitoring activity in each round. The participants performed significantly better with the blast and fade animations.

when using one type of animated display than another.

While the identification rate when monitoring information was not affected by device type, the time to react to it was. An examination of the monitoring times for the activities that were completed appears in Figure 10. The times to complete the awareness activities differed significantly depending on the type of animations used ($F(2, 52) = 17.24$, $MSE = 4528.75$, $p < .001$). Pairwise comparisons revealed the blast and fade animations resulted in significantly faster awareness times than the ticker ($p < .001$ and $p = .01$, respectively) and there was a trend toward faster blast times than fade ($p < .09$).

In examining the awareness responses, recall that each question had four possible answers, one or more of which were correct. Section 5.1 described three metrics for measuring performance on the awareness questions: the correctness rate, the hit rate,

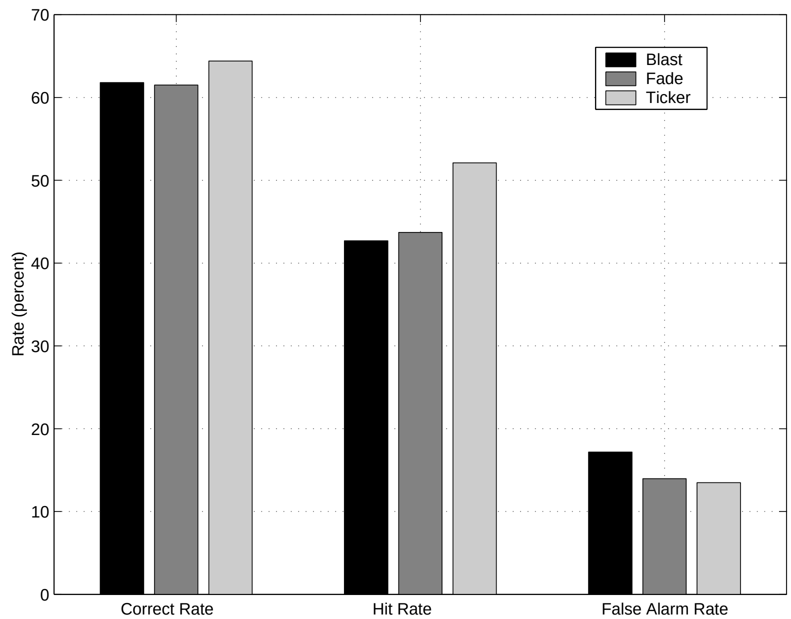

Figure 11: Cumulative correctness rate, hit rate, and false alarm rate for the awareness questions. The participants had a significantly higher hit rate when using the ticker.

and the false alarm rate. Figure 11 summarizes the results.

While the correctness rate for ticker (64.0%) was slightly higher than that for fade (58.3%) or blast (58.2%), suggesting that the ticker may be better, the difference was only marginally significant, $F(2,96) = 2.62$, $MSE = 0.02$, $p = .08$. In turning to the hit rate, there was a significant difference among hit rates, $F(2,96) = 3.87$, $MSE = 0.03$, $p < .03$. The hit rate for ticker (52.11%) was higher than that for fade (43.66%) and blast (42.72%).

Recall that one drawback of using techniques to achieve higher hit rates is that they often result in higher false alarm rates as well. However, this does not seem to be the case in this situation as there was not a significant difference between false alarm rates, $F(2,96) = 0.55$, $MSE = 0.03$, $p = .58$. The false alarm rate for blast (17.18%), fade (13.97%), and ticker (13.48%) are about the same.

The next section discusses the results.

5.2.3 Discussion

The result indicating that browsing times are not affected by the presence of animation may seem surprising, but a closer examination shows that it is not unreasonable. One might suspect that participants who are not faced with an animation and not burdened with the additional monitoring activities and awareness questions would perform significantly faster on the browsing tasks, but the results suggest that this is not the case. In fact, the results do not even indicate a trend toward lower times in the cases when an animated display is present. While this seems to contradict the results found in prior studies [45], recall that the primary task in those studies were editing tasks that required participants to perform in-depth readings and make corrections. The browsing tasks were less cognitively demanding, but based on the results of the pilot study in Section 4.1, they also seem to better match the type of primary tasks that a user would be doing while using a peripheral animated display.

The times to complete the monitoring activities differed significantly depending on the type of animations used. The blast and fade animations resulted in significantly faster monitoring times than the ticker. This result seems to follow from previous results that indicate that moving text is more difficult to read than non-moving text [67, 28]. As the tickering display relies on motion to cycle between items while the fade and blast do not, it seems reasonable that the ticker would result in slower performance, particularly if the participants were reading the displays to identify information that they were monitoring.

In analyzing the responses to the awareness questions, the correctness rates for the three animation types suggested that the ticker may be better, while the hit rates verified that the ticker resulted in significantly better performance. This does not contradict the results noted previously indicating that moving text was more difficult to read than non-moving text. The nature of the monitoring activities and the awareness questions are quite different. In fact, other studies have shown that comprehensibility, unlike reading speed, is not affected by motion [37], so it is reasonable to expect that the results would differ. The impact of these two results on the development and use of animated displays is clear: if the goal is to identify items quickly, an in-place display like a fade or blast should be used, while if the goal is to increase comprehension and memorability, a motion-based display like a ticker should be used.

5.3 Experiment 2

The previous experiment suggested that there are differences in performance when using the fade and ticker displays. In a followup experiment, I wanted to explore whether certain factors, namely display size and animation speed, impacted performance in any way. Perhaps making the display area larger would result in faster recognition times and allow the awareness questions to be answered with greater accuracy, or perhaps a slower speed would be less distracting, resulting in lower times on the browsing tasks.

5.3.1 Method

Ninety-one Georgia Tech undergraduate students participated in this experiment for class credit. The materials and procedure were similar to the ones used in the previous experiment with the differences described here.

A between-subjects size and speed condition was added. The participants were presented with a display having one of three characteristics: normal size and speed, normal size but slow speed, or small size but normal speed.

The *normal* displays were used as the comparison point for the small and slow displays. Normal displays used large display areas and fast speeds, though both well within the ranges of sizes and speeds selected by participants in the pilot study. Both the fade and ticker had a width of 1180 pixels (about 160 characters) with a height of one line. This size was chosen because it fits nicely along the top or bottom of the screen and because it is large enough to hold long streams of information (like news headlines and weather bulletins) in their entirety. The ticker speed was at the upper range of the possible speeds for the platform, with one-pixel updates every 20 milliseconds. Thus, for the 12-point font used in the experiment this translates to about 6.67 characters per second, meaning that a 20-character ball score or stock quote would require a little over three seconds to ticker on to the display, about 24 seconds to ticker across the 160-character display, and another three seconds to ticker off the display. During the experiment, the ticker experienced a slight slowdown that makes it difficult to compare the speeds of the ticker and fade. The fade cycle step had a 100 millisecond delay between each of five steps with a three-second delay before the next fade, resulting in 3.5 seconds elapsing in a fade cycle.

The *small* display used a smaller area but the same speed as the normal display. The fade and ticker width was more than halved to 840 pixels (about 70 characters), small enough to fit above a single window. This reduction in size meant that most streams of information could not be shown in their entirety using the fade widget.

Unlike in the previous experiment where long streams were truncated, in this experiment they were split over multiple screens. As such, it seemed that performance when using the fade widget would suffer dramatically.

The *slow* display was the same size as the normal display, but slower. The speed was chosen to be at the slow end of the range selected by participants in the pilot study. The ticker updated at a rate of one pixel every 140 milliseconds, seven times slower than for the normal display. The fade updated one shade every 150 milliseconds with a delay of 9 seconds before the next fade, resulting in 9.75 seconds between each cycle. The size for the widgets was the same as in the normal display.

The awareness questions also differed in style and number from the first experiment. Recall that in the first experiment, the first question asked participants to select the types of information that were displayed, then for each case where the participant stated correctly that a type of information was displayed, two additional questions were asked about that information, the first relating to content and the second relating to order. In this experiment, each participant answered all of the questions. This change seemed reasonable as a person's memory often can be sparked by seeing a word or phrase in a question. This also generated more data and made comparisons between participants more reliable.

The number of rounds was increased from six to eight. It was determined that participants would still be able to complete the experiment within the requested hour even with the additional rounds. Also, the type and order of the animations was changed. Since the blast display resulted in performance similar to the fade display and was consistently rated as the least favorite display by participants in the first experiment, it was not used in the second experiment. Participants alternated between using fade and ticker in each round, with one group starting with fade, then twice using ticker then fade again and repeating (fade, ticker, ticker, fade), and the other group reversing the order by starting with ticker (ticker, fade, fade, ticker).

In summary, there were six groups of participants differentiated by animation classification (normal, slow, or small) and starting animation (fade or ticker). Each group had 15 participants except the slow fade-first group with 16 participants.

5.3.2 Results

This section presents the performance results for the browsing tasks, monitoring tasks, and awareness questions.

For both the fade and the ticker, changes to the size and speed of the animated display did not lead to differences in the the cumulative time required to complete the browsing tasks, with $F(2,88) = 0.58$, $MSE = 765992.08$, $p = .56$ for fade, and $F(2,88) = 0.01$, $MSE = 997900.50$, $p = .98$ for ticker.

Does the size of the display affect performance on monitoring tasks? Figure 12 suggests that it does. When using the ticker, the time to complete the monitoring activities was significantly different based on the size of the animated display, $F(2,29) = 5.23$, $MSE = 40792.90$, $p = .01$ with $p = .02$ for the pairwise t-test. When using the fade, despite the large difference in the means, the monitoring time did not differ based on the size and speed of the animated display, $F(2,66) = 1.62$, $MSE = 12712.58$, $p = .21$.

Does the speed of the display affect performance on monitoring tasks? As seen in Figure 13, for each display type the monitoring times appear to have been similar regardless of speed. The analysis verifies that there was no significant difference: for

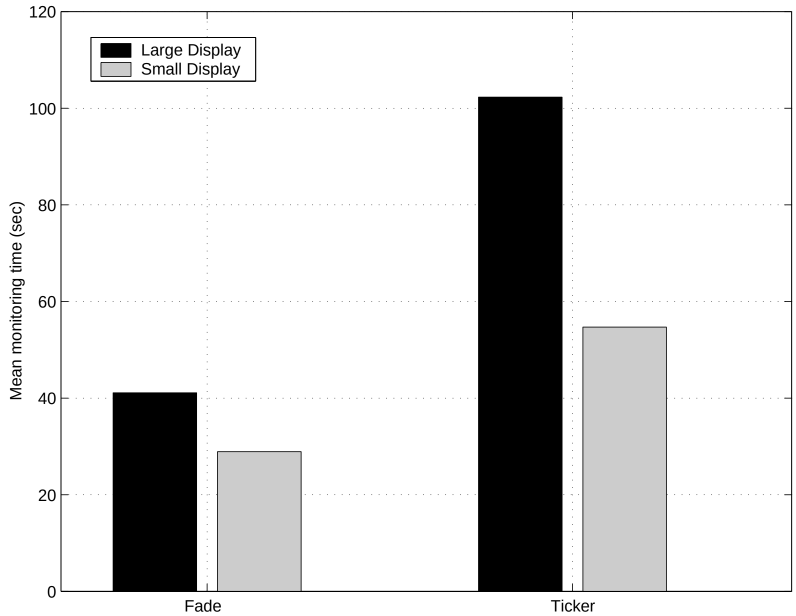

Figure 12: Mean completion times for each monitoring activity when using large and small displays. Smaller displays resulted in lower times than the larger "normal" displays for the monitoring activities.

the ticker, the t-test resulted in $p = .79$, for the fade, $p = .87$.

The cumulative hit rate when using the fade was virtually identical regardless of display size or speed, $F(2, 88) = 0.09$, $MSE = 0.02$, $p = .92$. However, when using the ticker, the hit rates did differ significantly, $F(2, 88) = 3.26$, $MSE = 0.02$, $p = .04$. As one might suspect from Figure 15, a t-test revealed that the difference between the normal and slow displays was significant ($p = .03$), while the difference between the normal and small displays was not ($p = .98$). The false alarm rate did not differ regardless of display size or speed for either the fade ($F(2, 88) = 2.24$, $MSE = 0.01$, $p = .17$) or the ticker ($F(2, 88) = 0.10$, $MSE = 0.01$, $p = .91$).

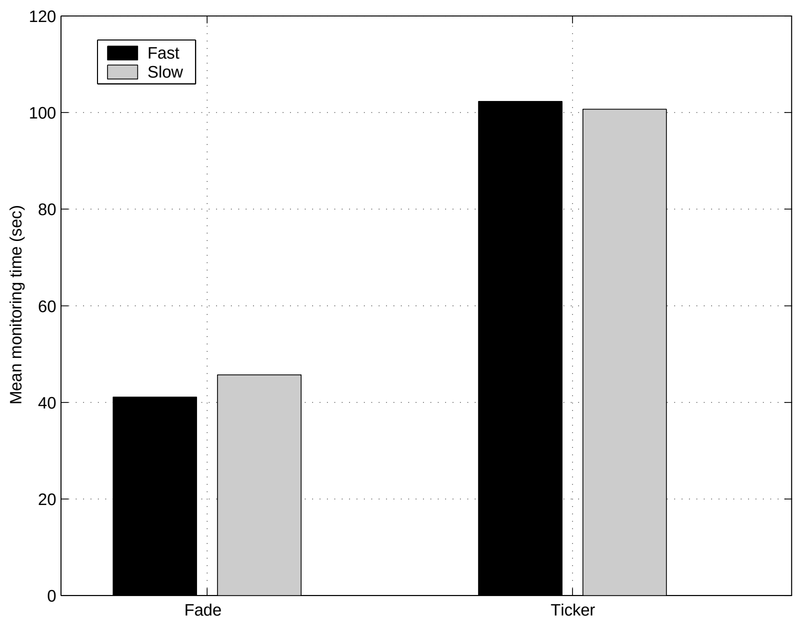

Figure 13: Mean completion times for each monitoring activity when using fast (normal) and slow displays. For each animation type, there was no significant difference in monitoring times based on the speed of the animation being used.

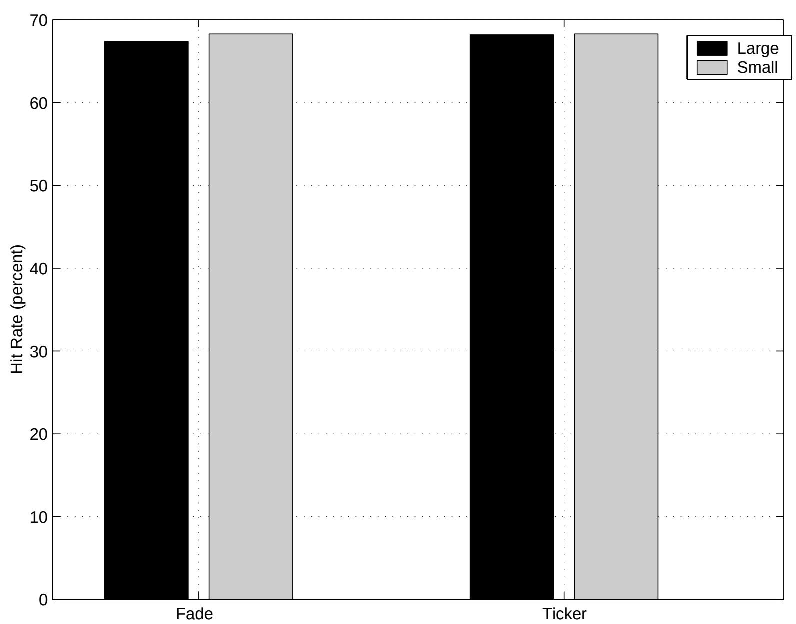

Figure 14: Cumulative hit rate for the awareness questions when using large and small displays. There is not a significant difference between the hit rates.

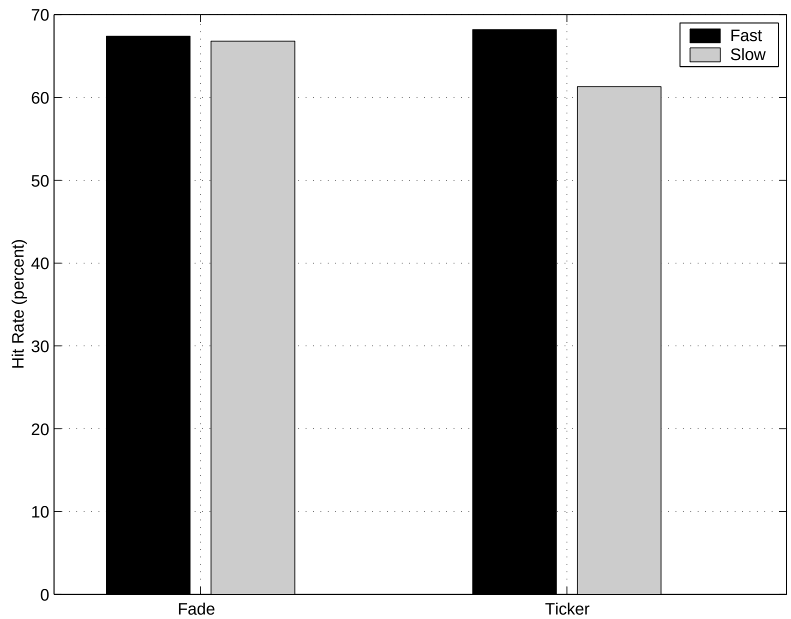

Figure 15: Cumulative hit rate for the awareness questions when using fast and slow displays. The slow ticker resulted in a significantly lower hit rate.

5.3.3 Discussion

While one might suspect that changes in the nature of the display might result in differences in browsing task times, this was not the case. This result extends the corresponding one from the previous experiment: as was noted previously for the type or presence of an animated display, neither the size of the display nor the speed of the animation seems to negatively impact the time required to complete browsing tasks.

A fast animation does not convey information more quickly than a slow one for either the fade or the ticker display. As seen in Figure 13, the response times for both types of displays were remarkably similar. One possible explanation for this result is that even though the increase in speed gives participants more opportunities to see the information, each opportunity is shorter because the information disappears more quickly. These two factors could balance out to result in similar monitoring times.

Although the speed of the display does not seem to affect monitoring times, the size does. A smaller display results in significantly faster monitoring times when using the ticker, and there is a trend toward faster times when using the fade (see Figure 12). While this result is not immediately intuitive, it does seem to correspond to the model that most people use during peripheral monitoring activities: they are focused on a primary task while occasionally glancing at the peripheral display. As noted by Rayner [64], only a limited number of characters (up to 20) can be processed in a quick glance at a display. The greater number of characters in the larger display, particularly for the ticker, may make it harder for a person to find the desired information with a quick glance.

While the size of the display seems to impact performance on the awareness activities, it does not seem to impact performance on the awareness questions that the participants answered at the end of each round. The participants performed equally well on the questions whether using the larger or smaller display. The reason why the display size makes a difference in answering the questions but not in performing the awareness activities lies in the nature of the two tasks. Whereas the awareness activities require only a quick glance, the ability to answer questions requires a more careful reading of the entire information entry. In fact, it is somewhat surprising that the larger display did not result in better results than the smaller one, but this may be because the smaller display was still large enough to contain most or all of the information for many of the information entries.

The speed of the tickering display seems to impact performance on awareness questions. A slower animation resulted in poorer performance on the questions than an faster one, perhaps because the participants too often glanced up to see the same information. By repeatedly glancing at the display and reading virtually the same information, they may have learned not to read and process the information very rigorously. This result follows from the Granaas work described in Chapter 2 that showed faster tickers (with larger jumps in the number of characters scrolled) resulted in better comprehension than slower tickers with smaller jumps [28].

In comparing performance on the awareness questions between experiments, note that the result from the previous experiment was not replicated: the ticker did not result in improved performance over the fade. This may be related to the amount of information on the screen. The tickers in this experiment were as large or larger than the ones in the first experiment, resulting in far more information on the screen at any given time for the ticker, especially compared to the fade. The shape of the display may also be a factor. Recall also that the fade display in the first experiment was a three-line display, while in the second it was a one-line display identical in size and shape to the ticker. It is possible that multi-line displays are more difficult to process and comprehend in a glance than single-line displays. Future work is necessary to test that hypothesis.

5.4 Post-experiment questionnaire results

While the timing results reflect how effectively potential users could maintain awareness using animated displays, it is also important to consider subjective preferences. If an individual performs slightly better at a task with a ticker but greatly prefers using a fade, then perhaps in all but extreme circumstances the fade display should be used.

To better understand subjective preferences, participants answered a series of questions about their perceived ease of use, intrusiveness, and potential future use of the different animations. The questions were as follows:

```
1. How easy was it to keep track of the information in the display?
   (-2 = not easy, 2 = easy)
2. How intrusive to your browsing were the updates to the display?
   (-2 = not intrusive, 2 = very intrusive)
3. How often would you use a program that employs the display?
   (-2 = never, 2 = all the time)
```

At the end of the first experiment, participants were asked the series of three questions for each of the three display types for a total of nine questions. The results are presented in Table 7.

In the post-experiment responses for the first experiment, the blast was rated least easy to use and most intrusive, and the ticker was rated easiest to use and least intrusive. Thus, even though the blast display often resulted in performance as good or better than the other displays, it may not necessarily be the best choice since users did not seem to enjoy using it. However, recall from the experiment results that the participants did not perform any more slowly on the browsing tasks when using the blast display than when using the fade or ticker. People seem able to adapt and perform equally well despite their dislike of the blast display.

At the end of the second experiment, the participants were again asked the series of questions about their preferences for animation, this time for the two display types

Ease of Use

	-2	-1	0	1	2	Mean
Fade	11	19	10	10	4	-0.43
Ticker	8	14	7	16	9	0.07
Blast	15	23	8	8	1	-0.78

Intrusiveness

	-2	-1	0	1	2	Mean
Fade	9	13	15	13	5	-0.15
Ticker	14	18	12	10	1	-0.62
Blast	6	11	11	18	9	0.24

Future Use

	-2	-1	0	1	2	Mean
Fade	11	16	13	11	4	-0.34
Ticker	3	7	16	24	5	0.38
Blast	16	16	12	10	1	-0.65

Table 7: Result totals from the first experiment's post-experiment questionnaire for the three questions. (Two no-answers for the first question result in uneven sums.) -2 indicates the most negative response to the question, 2 the most positive.

used by each participant: fade and ticker. Recall that there were three groups of users: the normal group that used a normal size and speed, the slow group that used a slow speed but normal size display, and the small group that used a small size but normal speed display. Table 8 summarizes the participants' responses to the three questions.

For the fade display, the participants tended to produce negative responses for ease of use, with 57 negative responses compared to only 14 positive ones. The smaller and slower displays resulted in more negative responses than the larger and faster one. The most negative responses for ease of use were shown with the small display with over 5 times more negative responses than positive ones. No doubt this is because information is often split over multiple screens in the small display.

In comparison, participants found the ticker relatively easy to use, with positive responses outranking negative ones by a 3.5 to 1 margin for normal size and speed and a 2 to 1 margin for the slower speed. The smaller size had a 1 to 1 ratio of positive and negative responses. Since participants only used one size of display, they were not biased by having used a larger or faster display but rather they genuinely seemed to prefer the larger and faster display.

While participants found the larger and faster display easier to use, they did not think it was significantly more intrusive. The positive and negative responses were fairly comparable across sizes and speeds for the intrusiveness ratings. However, while participants found the ticker easy to use, they also found it to be somewhat intrusive, with almost a third more negative responses than positive. Again, the small display had the most negative responses, with 50 percent more negative than positive responses.

Ease of Use (Fade)

	-2	-1	0	1	2	Mean
Normal	1	15	8	6	0	-0.37
Slow	7	12	7	4	0	-0.73
Small	10	12	5	3	1	-0.87
All	18	39	20	13	1	-0.66

Ease of Use (Ticker)

	-2	-1	0	1	2	Mean
Normal	1	4	6	13	5	0.59
Slow	1	8	5	11	5	0.37
Small	3	10	4	10	4	0.06
All	5	22	15	34	14	0.33

Intrusiveness (Fade)

	-2	-1	0	1	2	Mean
Normal	3	5	9	8	5	0.23
Slow	2	5	9	12	1	0.17
Small	1	5	10	12	2	0.30
All	6	15	28	32	8	0.24

Intrusiveness (Ticker)

	-2	-1	0	1	2	Mean
Normal	4	8	8	9	1	-0.17
Slow	5	8	5	9	1	-0.25
Small	3	9	11	6	2	-0.16
All	12	22	24	24	4	-0.23

Future Use (Fade)

	-2	-1	0	1	2	Mean
Normal	7	8	8	6	1	-0.47
Slow	4	12	11	3	0	-0.43
Small	11	7	9	4	0	-0.81
All	22	27	28	13	1	-0.61

Future Use (Ticker)

	-2	-1	0	1	2	Mean
Normal	1	5	8	14	2	0.37
Slow	2	6	8	13	1	0.17
Small	4	4	12	8	3	0.06
All	7	15	28	35	6	0.20

Table 8: Participant responses to post-experiment questions for the second experiment regarding the ease of use, intrusiveness, and predicted frequency of future use for the fade and ticker animated displays. -2 indicates the most negative response to the question, 2 the most positive.

For the fade display, the most frequently selected response for frequency of future use was 0 and -1, though almost a quarter of the participants indicated that they would never use it. The most significant contributor to the negative response was the small display. Even though participants found the ticker to be fairly intrusive, most indicated that they would use it often or always. Positive responses outpolled negative ones by a almost a 2 to 1 margin. Even the small display, which participants found least easy to use and most intrusive, generated almost 50 percent more positive responses than negative.

In summary, the participants found the ticker easier to use than the fade. While participants thought the ticker was easier to use, they also found it to be more intrusive. Even though participants found the ticker more intrusive, they indicated that they would be more willing to use it again than they would the fade.

The participants did not seem at all anxious to use the fade or the ticker at all times, though few people ruled out the possibility that they would ever use it. This indicates that more applications should be targeted for short term use, like ball score monitors that are only used for the duration of a game or traffic monitors used only in the hour before leaving to drive home.

5.5 General discussion

The goal of the empirical evaluations was to explore the balance between distraction, reaction, and comprehension and memorability when using peripheral animated displays. The first experiment showed that fading, tickering, and blasting peripheral displays did not significantly distract users from a primary task yet can effectively communicate information, and that the type of animation impacts performance on awareness tasks. The second experiment showed that changes in size and speed can impact performance on awareness tasks.

The following recommendations can be derived from the results of these experiments:

- **Animated displays can be used in the periphery with minimal negative impact on certain primary tasks.** Both experiments supported this claim when the primary task involves browsing for numerical solutions in a hypertext environment.

- **In-place displays such as fade and blast are better than motion-based displays like ticker for rapid identification of items.** Participants were able to complete monitoring activities more quickly when using the fade and blast than when using the ticker. This seems to extend prior results that indicated that moving text is more difficult to read than static text.

- **Motion-based displays such as ticker are better than in-place animations for comprehension and memorability.** Participants who used the ticker obtained a better hit rate than those who used the blast and the fade.

- **Small displays result in faster identification of changing information.** This may be related to the amount of information that a viewer can read in a single glance. Larger displays may make it difficult to obtain the right information.

- **Fast displays are better than slow for establishing comprehension**

and memorability. This was noted in the second experiment for the tickering animation and may be related to the amount of new information that is available in a single glance.

One frequently-mentioned limitation to empirical evaluations is that they often do not represent an accurate representation of real-world situations. For example, in the experiments, participants used the animated display only for up to an hour. However, there are many real-world situations where this would be a reasonable time frame, perhaps to monitor the traffic between 5 pm and 6 pm every weekday, or to keep an eye on the scores of selected baseball games during the pennant drive and playoffs. In fact, it seems advisable that programmers should not write applications that employ animation with the expectation that they will be used continually, but rather for short, well-defined periods of time.

Another potential limitation of the study is that the population selected to perform the studies consisted of undergraduate students attending a technical school. However, this is perhaps the ideal group to examine if one hopes to obtain positive results. They are part of the computer and video-game generation that has been bombarded with animated displays their entire lives. It is unclear how well animated displays would work for older adults who have poorer physical skills and less computer experience, or very young children who have never used a computer, or even individual segments of the university population. As a first step in understanding how user populations differ, the next chapter describes an observational study that characterizes users based on their animation preferences and considers ways to identify them using readily available habits and traits.

CHAPTER 6
Supporting Awareness with Animated Widgets

In designing and evaluating peripheral applications for supporting awareness, I considered many existing toolkits and languages for generating the animated effects. While there are many choices for adding animation to interfaces, few provide easy support for maintaining awareness. The constant, cyclic effects shown to be effective in the previous chapters are difficult to achieve in existing toolkits, which instead focus on providing very general support for changing some characteristic of an object (position, color, visibility) in a fixed period of time. In addition, I wanted to consider methods for raising awareness of additions and changes to the information being displayed, something that is not at all addressed in other toolkits.

To provide easy creation and use of constant, cyclic animated effects, I have integrated animation support into the Tcl/Tk scripting language [58] via three widgets. I focused on three effects, two motion-based effects and one in-place effect. The *fade* widget cycles between items of text, bitmaps, or images; the *ticker* widget scrolls information horizontally across the screen; and the *roll* widget scrolls information vertically. Section 6.1 describes the widgets in more detail.

In creating an animated widget set, numerous design decisions had to be made, not the least of which is the choice of a language. The Web-aware nature of Tcl combined with its powerful string parsing capabilities makes it a good choice for information monitoring applications. The easily extensible widget set and well-defined programming interface of the Tk toolkit provide a good framework for the implementation of animated widgets. Section 6.3 outlines selected implementation details made in constructing animated widgets, including more reasons behind the choice of Tcl/Tk as the implementation language.

While animation has many potential benefits, it has several drawbacks as well. Despite evidence to the contrary in Chapter 5, animation is often seen as distracting or annoying because of its constant motion and rapid visual changes. The widget set addresses this problem by giving both the programmer and the end-user significant power over the animation. Another problem with animation specific to awareness is that it can be difficult to see where changes occur and what the previous states of information was. By augmenting the display using information from previous states with techniques I call *automatic highlighting* and *history-based shadowing*, I attempt to lessen the effects of these problems. Section 6.4 describes in detail the solutions.

The animated widgets have been available and in use for over a year. They have been used by over fifty programmers in a variety of information awareness applications, and some of the programs have been used by more than a hundred people. Section 6.5 discusses some guidelines for writing information awareness applications based on feedback from both programmers and users.

Overall, I believe that constant, cyclic animation is an important technique to have available in a user interface toolkit to simplify the building of awareness applications. This chapter describes the behavior of the Agentk animated widgets and explores some of the design decisions made in creating them.

6.1 Animated widgets in the Agentk toolkit

Animation has the potential to enhance the awareness capabilities of interfaces. The Agentk toolkit contains several interface widgets intended to make it easy for programmers to design awareness applications and easy for people to use them in maintaining awareness of changing information.

Animated widgets are a subclass of *mega-widgets*, a collection of widgets that are operated using a single interface. For example, the LabelText mega-widget in the Tix library provides a single interface to a label widget and an entry widget [40]. Just as mega-widgets augment the power and ease of use of widgets by packaging several of them together as a single new widget, animated widgets augment the behavior of a widget by constantly changing its appearance at regular intervals.

Agentk is a widget toolkit designed to assist in the creation of agent-like programs such as the information awareness applications discussed earlier. Agentk contains three animated widgets: the *fade* widget, the *ticker* widget and the *roll* widget. The fade widget fades between multiple blocks of text, bitmaps, or graphical images. It can be used when the blocks of information have a fixed height and width. The ticker widget horizontally scrolls or "tickers" a stream of text. It can be used for variable length streams of text. The roll widget scrolls text vertically. It is well suited for ordered lists of items.

The programming interface for the fade, ticker, and roll widgets is identical to that for labels, scrollbars, or any other widget: the programmer specifies the widget creation command (ex. label, scrollbar, fade), the position in the display tree (.fader, .dialog.ok), and possibly some options (-width 50, -fg red, -shadowhistory yes). The

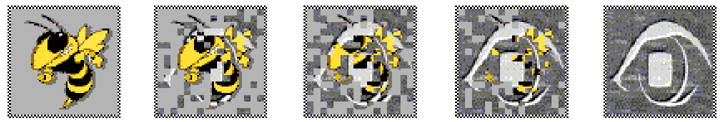

Figure 16: A time-lapse series of the fade widget for two images. Rather than perform compute-intensive calculations to achieve a fading effect, the original image is broken into pieces, and the pieces of the original are gradually replaced with pieces of the final.

options include all those of the standard label widget used to display static text and images, as well as additional options that allow a programmer to show the contents of more than one variable and to control aspects of the animation. In so doing, a programming interface that can be easily understood by a Tcl/Tk programmer is provided.

The remainder of this section briefly introduces the widgets and some of the available options. A complete list and description is available in Appendix B.

6.1.1 Fade widget

Rapid changes in the appearance of a widget often attracts the user's attention to the widget. While this is advantageous in many situations, it could be detrimental in an interface where the users' attention is primarily focused on other tasks. Instead, a widget is needed that can change continuously to match the large and dynamic information space but will change gradually to avoid interrupting the everyday tasks of the user.

The fade widget displays multiple blocks of either text, bitmaps, or graphical images within a given space by gradually fading between them. The gradual change should be less distracting than a sudden switch yet will allow multiple information blocks to be displayed in a single area. The speed with which the fade occurs can vary depending on the nature of the application: if the widget is designed to attract attention and can be easily stopped by the user, a quick fade might be used, while a secondary display designed to run all the time might call for a slower, less intrusive fade. Not only can the fading speed be controlled, but also the time between fades. Many of the observational study participants from Chapter 7 indicated that for textual information they wanted a rapid fading speed but a long duration between fades to give them time to read information but to minimize the time in a transitional state when it is difficult to read the information.

The following lines of code give an example of the fade widget displayed in Figure 16.

```
set i1 [image create photo -file buzz.gif]                        (1)
set i2 [image create photo -file eye.gif]                         (2)
fade .f -width 100 -height 100 -imagevariable [list i1 i2]        (3)
pack .f                                                           (4)
.f run                                                            (5)
```

The first two lines create two images that are to be faded (actually stippled) in and out. If the values of the variables are changed later, the new images will fade in on the next iteration. The third line creates the widget to display the information contained in the variables i1 and i2. Unlike the textvariable option for other widgets, the fade widget supports a list of variables, fading between each in the list.

Figure 17: A news agent that incorporates a fading widget (top right) and tickering widget (bottom right) to alert the user of new stories.

6.1.2 Ticker widget

The ticker widget provides a ticker-tape-style display that scrolls or "tickers" text across the screen. As with the fade widget, a gradual tickering can be less distracting than a sudden switch. The ticker widget has the added advantage that a stream of text could be any length since it never needs to be displayed in its entirety at any given time.

In addition, it takes advantage of the natural reaction to grab and pull the ticker widget to make it slow down, stop, back up, or move. When users click and hold down the mouse button within the widget, they can "throw" the widget by dragging it in a certain direction at a desired speed, thus causing the information to move in the indicated direction and speed. By providing this functionality, the user has more control over the way in which the information is displayed.

The following lines of code gives an example of the ticker widget.

```
ticker .t -width 100 -textvariable [list u1 u2 u3]          (1)
pack .t                                                      (2)
.t run                                                       (3)
```

The first line creates a 100-character-wide widget that is configured to display the information in the variables u1, u2, and u3. The second line packs the ticker widget just as any other widget would be packed, and the third line begins the tickering process.

The ticker widgets have been used in applications to show article headers which, unlike sports scores or stock quotes, can vary widely in length. Figure 17 shows an interface where the ticker widget has been used.

6.1.3 Roll widget

The roll widget "rolls" or vertically scrolls text across the screen. It seems best suited for a list of items or a columns of information where the order and position of the information is important. The ordering of the list and the positioning of the columns is more apparent in the roll widget than in a ticker or fade widget. As with the ticker widget, the user can grab, hold, and move the roll widget to alter the visible information.

Consider the following modest example for monitoring a printer queue on a Unix system.

```
roll .r -width 60 -height 6 -justify left -font fixed      (1)
pack .r                                                    (2)
.r run                                                     (3)
proc checkq {} {                                           (4)
  .r configure -text [exec lpq]                            (5)
  after 20000 checkq                                       (6)
}                                                          (7)
checkq                                                     (8)
```

The first line creates a roll widget that is six lines high and 60 characters wide. The

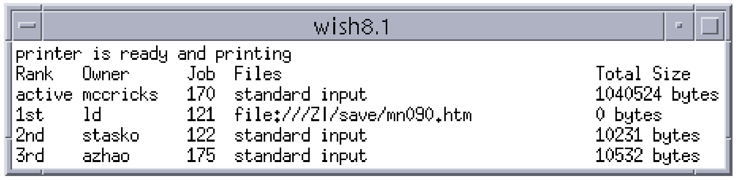

Figure 18: A roll widget that displays the contents of a printer queue. The information is rolled vertically across the screen. The user can grab and move the display if desired and can throw (drag and release) it to adjust the speed.

text is left justified and the font is fixed to ensure that the columns will line up. The second and third lines pack the widget and start it running. The checkq procedure repeatedly checks the contents of the printer queue using the Unix lpq command and configures the roll widget to reflect the results of this command. Figure 18 shows the result of this script.

System administrators could use this short script to keep an eye on buggy printers or to watch for printer misuse. A user who is printing many documents over a short period of time could extend the script to watch several printers at once to know at any time which is the least busy.

One important consideration in designing a vertically scrolling widget was whether to include scrolling direction as an option in the ticker widget or create a separate widget. It was decided that creating a separate widget would help lessen confusion and facilitate good design. Since the options for ticker and roll differ somewhat (see Appendix B), a single widget would have options that at times are not valid. Also, by separating the widgets, designers have to make a decision in advance as to the animation type best suited for the target application, which should promote better application design.

6.2 Applications using animated widgets

The animated widgets in Agentk have been used in a variety of programs by over fifty programmers. The figures in this chapter show some of the interfaces they have designed. This section focuses on a few interfaces, all of which are available from the Agentk home page.

The NewsAgent interface repeatedly downloads and parses a news Web page looking for new articles (see Figure 17). It alerts users of new articles by changing the color of an image and rapidly fading in and out a "New News" message. The interface contains a ticker widget that continually tickers through the news headers. If a user sees an article of interest, by pressing the image a news viewer can be raised and the user can read the article in its entirety.

The tkscore interface monitors college basketball scores and displays them using a user-selected animated widget or using a standard label widget. The pilot study described in Section 4.1 using this interface has been used to further the development of the animated widgets and have led to the development of the tkwatch interface, a more general tool for monitoring stock quotes, news headlines, weather data, personal information, and sports scores.

The CWIC passive browsing system (see Figure 21) continually browses selected Web sites, identifying key images and presenting them to the user by gradually fading between images. Note that the URL where each image was found is provided as well so that the user can visit the page if an image of interest appears. CWIC is described in detail in [9].

Agentk has been used outside of Georgia Tech as well, where one project in particular stands out. Raphael Finkel of the University of Kentucky integrated Agentk into Pulsar, a system for system administrators designed to monitor machines, network connections, printers, and other shared resources. In Pulsar, information falls into one of three categories depending on its importance: green, yellow, or red. In its original form, all red alerts caused a button to change color, and the user had to navigate through a series of menus to find the source of the alert. Pulsar was augmented so that all red alerts could be tickered or faded on the screen. When users see the ticker or fade, they are provided with enough information to determine whether the event requires immediate attention. If the situation worsens, this information is also available. The animated display also provides a constant reminder of the alert, so the user is reminded to deal with the situation when there is an opportunity to do so.

6.3 Incorporating animation into Tcl/Tk widgets

The style of programming represented by scripting languages is well-suited for information awareness applications. As noted by Tcl creator John Ousterhout, scripting languages are designed to "glue" together existing resources making them easier and more efficient to use [59]. As this closely matches the purpose of awareness applications, (to act as an intermediary to the monitoring of information resources that otherwise may be tedious and time-consuming to do), it seems that scripting languages are an appropriate choice for authoring awareness applications. Tcl/Tk in particular seems to be a good choice for awareness applications. Tcl is designed to be a platform-independent scripting language with an extensible graphical toolkit (Tk) for interface design [58]. Extending this toolkit to include animated widgets provides a solid programming platform for the implementation of awareness applications.

To maintain consistency and decrease the learning time for programmers, animated widgets behave like standard Tk widgets. To simplify this process, the Agentk animated widgets use the Tk requirement library for widget creation, querying, and modification (see http://www.hobbs.wservice.com/tcl/script/widget/) created by Jeffrey Hobbs. The widget package provides a framework for selecting components, creating subcommands and options, and integrating related procedures into a single namespace. Each of the animated widgets has as its sole component the canvas widget. This means that even though the animated widgets appear to behave like a standard label widget, they can make full use of the additional display capabilities of the canvas widget. Initial implementations used a label, but it seems that the canvas was necessary to provide all of the desired behaviors.

The Agentk widgets are implemented in Tcl only, meaning that they can be used on any platform with no compilation. The entire package consists of about 4000 lines of code (plus an additional 10000 for the widget package), and individual widgets can be included or excluded as needed to further reduce the overhead. The widgets have been tested on various Unix platforms as well as Windows 95 and 98. All of the widgets can take advantage of the additional image formats and capabilities found in Jan Nijtman's Img extension (see http://purl.oclc.org/net/nijtmans/img.html) but it is not necessary to have the extension to use Agentk.

The remainder of this section focuses on two key issues addressed in the creation of the animated widgets in Agentk. The first addresses performance across platforms, particularly important given that slower performance for animations results not simply in longer delays, but in a different appearance.

6.3.1 Maintaining platform independence

The loop that creates the animation effect (similar for each of the widgets) can be summarized as follows:

```
proc anim {} {
  # calculate next animation step
  ...

  after $delay anim
}
```

The after command waits for the period of time specified in the delay variable before executing the anim command.

Initial Agentk implementations ignored the time required to perform the calculations. On a slower or more heavily-loaded machine, the calculations may take significantly longer, resulting in an animation that is slower. Thus, a program may look very different depending on the platform. To address this problem, the widgets calculate the required to complete the calculations and subtract it from the delay. If the resulting delay was less than zero, the size of the steps taken by the algorithm was adjusted. For example, a ticker might move two pixels instead of one, or a fading of text might make larger changes in color over a smaller number of steps. This is similar to the approach used in prior automatic tuning [34, 75].

By taking this approach, the animated widgets will look similar on any platform, regardless of the processor speed or machine load. The only perceivable difference is that the animation may be smoother on faster, less-heavily-loaded machines. By allowing for this automatic adjustment in performance, a programmer can write a script using animated widgets with confidence that it will appear the same to users on a variety of platforms.

6.3.2 Calculating an animation step

In fading between blocks of text or bitmaps (see the lower portion of Figure 21), the foreground color of the original information is gradually changed to match the background color, in essence fading the information away. At the same time, the new information (originally "invisible" because it starts as the background color) changes to match the foreground color. When the new information becomes closer (in an RGB-value sense) to the foreground color than the original information, it is raised to the top of the display stack.

When fading between two images (see Figures 16 and 21), it is impractical to change each pixel from the old color to the new — even a small 100x100-pixel image contains 10,000 pixels to repeatedly fade. Instead, the fade widget uses a stippling effect. Each image is divided into small squares, and the squares from the original are replaced with the squares from the new until the effect is complete. The user can specify the size of the squares or can specify the speed with which the animation will occur. If the size is specified, the speed will be dependent on the speed of the machine. A faster machine will be able to calculate and update the display more quickly than a slower one. If the speed is specified, the size of the squares will depend on the speed of the machine. Slower machines divide the images into larger squares, but the time required to fade from one image to another will be the same.

The tickering (and rolling) effects are accomplished by moving all of the items on the canvas horizontally (or vertically). As old textual and graphical items disappear from one side of the canvas, new items are added to the other. The number of pixels by which the display is shifted depends on the response time of the machine. Typically the display is shifted by a single pixel at each step, but slower machines may not be able to keep up with the desired speed at that rate. The widgets regularly monitor the performance and increase the number of pixels if the system is being taxed. Note that this will result in an animation that is not as smooth, but it is more important that the appearance be similar on all machines.

6.4 Compensating for animation drawbacks

This chapter has discussed ways that programmers can use the Agentk toolkit to incorporate animation into their programs in a platform-independent and resource-friendly manner. However, in creating a toolkit it is most important to consider the ability of the user to obtain the desired information. This section discusses ways that the Agentk animated widgets deal with three user-related concerns.

6.4.1 Highlighting changes with automatic markups

Although animation provides a means to smoothly show the current state of changing information, it can be difficult for a user to identify when and where a change has occurred. To address this problem, the Agentk animated widgets support automatic markups of text. These markups include boldface, italics, and coloring and can occur whenever the information in the widget is updated. For example, Figure 19 shows

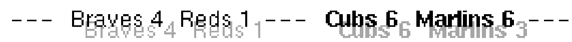

Figure 19: An example of automatic markups and history-based shadowing in a ticker widget showing sports scores. The bold text indicates a score that was recently updated, while the older score appears in plain text. The background shadow shows scores from ten minutes ago.

a widget that displays sports scores that are constantly downloaded from the Web. Recently changed scores are shown in bold text, while older scores appear in plain text.

A protocol is needed for removing the markups from older items. Agentk makes two options available to programmers: an automatic markup can be removed after a given period of time or after a given number of iterations of the display. In addition, the programmer can allow the user to reset the markups with a button press, mouse click, or other action. In the sports scores example, new scores could be highlighted for five minutes, or for ten iterations of the display. The programmer selects the option that seems best suited for the application.

Markups have long been established as a good way to highlight important parts of information. Systems like HtmlDiff [17] have used them to draw attention to changed information. The Agentk toolkit simplifies the integration of markups into situations where it is important to highlight changes. They are instantiated using command options that specify the desired style of markup and when they should be added and removed.

6.4.2 Showing previous states with history-based shadowing

Sometimes it is not sufficient to simply relate to a user that a change has occurred – it is necessary to communicate information about the nature of the change. To provide this capability, the widgets support *history-based shadowing*, a technique where a previous state of text is shown in the shadow of the current information state (see Figure 19). This technique captures the spirit of integrating supporting material described in [13] framed with a familiar shadow metaphor. In history-based shadowing, the shadow appears slightly offset horizontally and vertically from the original text and appears in a color that is between (in the RGB-value sense) the foreground and background colors. As such, it requires little extra space and is less obvious on first glance than the (more important) current state.

Similar to the automatic markups, the history-based shadowing for a given piece of information appears as soon as changes to the information are noted and can persist for a given length of time, a given number of iterations, or until the next change occurs. This gives the programmer the flexibility to choose a level of persistence that is best suited for the information.

History-based shadowing seems to be a good complement to animation and markups in maintaining awareness of information. The presence and persistence of the shadows is instantiated using Tcl-style command options. Shadowing provides a way to show not only the presence of changes to information, but also the state of the information prior to the changes.

6.5 Programming guidelines for animated widgets

One of the basic principles of interface design is to help the user maintain a sense of control over the actions on the screen. A user should not have to respond to actions but instead should control the rate at which information is assimilated. Yet animation seems to violate this principle: the flow of information typically is not under the direct control of the user. While the implementation attempts to empower the user by providing bindings for flow control, a programmer must be sensitive to the needs and desires of the user population. In fact, one criticism of animation in general is that it can be disruptive. The "blink" tag in HTML and animated advertisements on Web pages are often appropriately characterized as being annoying and disruptive. However, people have long used computer desktop accessories such as clocks and machine load monitors. What are the distinctions between these situations?

One distinction is that the burden of tolerating the constant changes is outweighed by the advantages these tools provide. Glancing at a desktop clock to obtain the time is far easier than running the date command or even looking at a wristwatch, and looking at a load monitor while running a compute-intensive program is far easier than running commands to determine the system load. A programmer should target application domains where the potential for knowledge is significant and the ability to start and stop the application is easy and apparent.

Another distinction is that the changes to the display are small, subtle, and predictable, allowing the user to adapt to the changing display to the point where it is barely noticeable. I hope that by providing smooth animations (many with user-controllable speeds), the applications will be less distracting. The programmer can assist in this by choosing appropriate sizes and speeds for the application. Users are more likely to use an application if it does not consume much desktop space and if it is not overly distracting.

Experiences indicate that users are willing to use applications that employ animated widgets, though many of them do not use them continually. I regularly receive comments from users who started up an information awareness application for a few hours, whether it be to keep track of the score of a game in the heat of the playoffs or to watch for new news about the JFK Jr crash. Thus, programmers should not necessarily write peripheral animation applications with the expectation that they will be used continually, but rather for short, well-defined periods of time, perhaps to monitor the traffic 5 and 6 pm every weekday, or to keep an eye on the scores of selected baseball games during the pennant drive and playoffs. If programmers do anticipate that it will be necessary to run the application at all times, alternate (non-animated) information delivery mechanisms should be made available.

Integrating animation into the Tcl/Tk toolkit and combining it with related highlighting techniques makes it possible for programmers to include it in their applications using a familiar programming interface. This makes it easier for users to keep a sense of control while maintaining a desired level of information awareness. Planned widget set extensions that will include other behaviors such as shrinking, growing, and swiping should further extend the user base. Adding other optional techniques such as motion trails and slow-in/slow-out should lessen the distraction caused by the animations as demonstrated in [41, 14, 34].

Programmer and user reaction to animated widgets has been encouraging (see Chapter 4), and the use should continue as the widget capabilities increase. The Agentk toolkit, including the animated widgets and most of the programs described in this thesis, is available at http://www.cc.gatech.edu/~mccricks/agentk. This site contains color screenshots of systems designed using Agentk, in-depth descriptions of the widgets, links to the latest Agentk release, and links to the papers and articles that discuss Agentk and its uses [47, 53, 48].

CHAPTER 7
Observational Study

The empirical evaluations in Chapter 5 examined whether features of an animated display could impact information acquisition during the awareness process. The observational study described in this chapter examines whether characteristics of potential users might also impact information acquisition. Though similar to the pilot study from Section 4.1, the information gained from the pilot study and the empirical evaluations made it possible to construct more rigorous questions and monitoring techniques for the participants. The types of available information were expanded to include suggested favorites, behaviors that were reported in earlier studies were monitored in this one, and in-depth questions were asked on issues that earlier studies only touched on.

This study focused on understanding user preferences for animated displays. The goal was to be able to understand the different ways that people might use peripheral animated displays. The approach described here attempted to learn personal characteristics and animation preferences of potential users and to uncover relationships between the characteristics and preferences. The expectation in this approach is that system designers should be able to predict animation preferences using readily available characteristics about the users. By better targeting applications to the needs of the users, designers will be able to create peripheral displays that allow their users to obtain information more easily.

7.1 The tkwatch interface

The tkwatch interface was expanded from the tkscore basketball monitoring interface described in Section 4.1. While tkscore only displayed scores for NCAA tournament games, tkwatch displays scores for a variety of sports events as well as weather data, stock quotes, and news headlines. The tkwatch interface allows a user to select or enter items to monitor. Figure 20a shows the interface for changing the monitored information.

As with tkscore, tkwatch users can select a fade, ticker, or list display. In addition, the appearance of the tkwatch displays are configurable. Users can alter the font, color, and speed of a display and can turn on or off the automatic highlighting and shadow history techniques described in Section 6.4. Figure 20b shows the tkwatch display interface.

For the observational study, the tkwatch interface was augmented to record data about the information that participants monitored and the manner in which it was displayed. Every action by a user to tkwatch, from selections of information and display options to clicks and grabs of the animation, was recorded. The data were stored in a `.tkwatch` subdirectory in each participant's home directory.

7.2 Study overview

To understand the impact of individual characteristics on the ways that people might use peripheral animated displays, I designed an observational study to collect personal data on individuals and usage data on the ways they use tkwatch. Participants willing to use tkwatch for an extended period of time were recruited from the Georgia

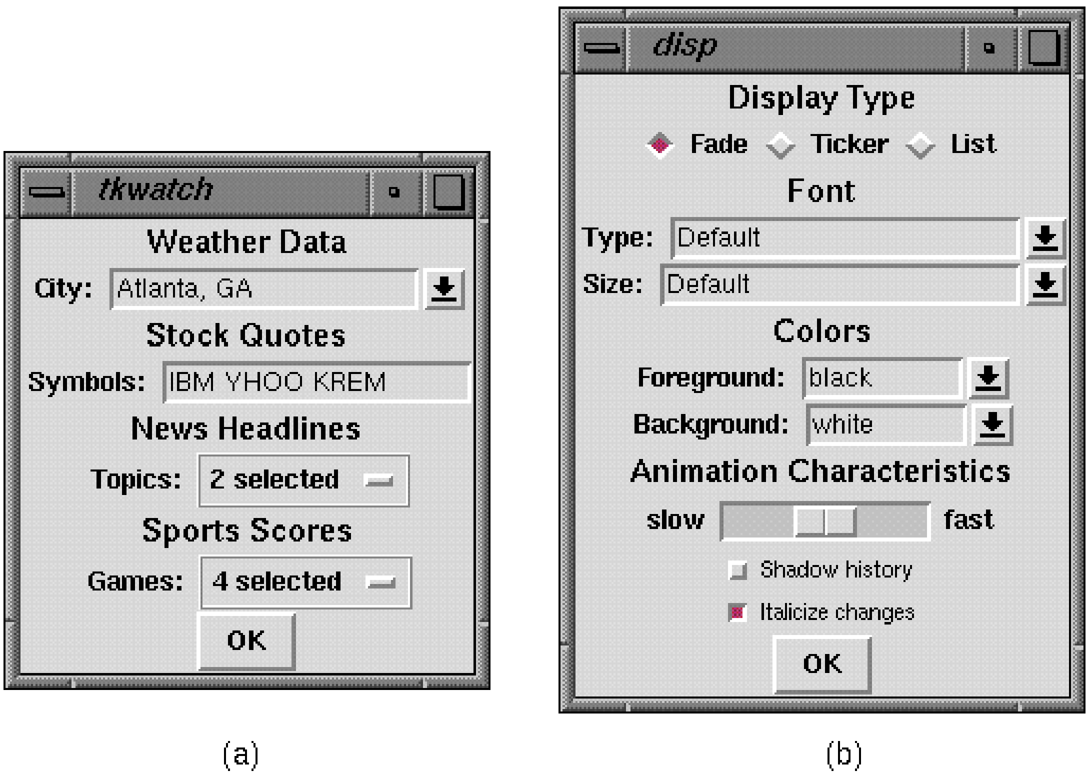

Figure 20: The tkwatch (a) monitoring and (b) display configuration interfaces. The monitoring interface allows users to select information to monitor, including weather for a city, symbols for stock quotes, categories of news headlines, and scores of games. The display interface lets users change various aspects of the display, including the animation type, the font type and size, the foreground and background colors, the animation speed, and the shadow history and automatic markup change indicators.

Tech campus. The participants were asked to complete a pre-study questionnaire that detailed their computer usage and information consumption habits. Then, they used tkwatch for several days, with all of their inputs recorded. Finally, one-on-one interviews were conducted to explore the reasons for their information monitoring and display choices.

A general call for participation was made to faculty and student members of the Georgia Tech College of Computing. A special invitation was made to participants in the tkscore pilot study and to members of the Information Interfaces group in the College. While a group that is knowledgeable about interfaces may not accurately reflect the population as a whole, these people seemed more likely to explore various configurations of the interface. In addition, the nature of their work provides them with the opportunity to use the interface extensively in a short period of time.

After agreeing to participate, each person completed a pre-study questionnaire focusing on two areas: computer usage and information consumption habits. The complete survey can be viewed in Appendix C; a brief summary follows here.

The computer usage questions asked for estimates of the number of hours per day spent at the computer, the percentage of time spent on various activities (coding, writing, email, Web browsing, etc.), and the types of desktop information monitoring tools (email biffs, clocks, load monitors, etc.) that the participants use. These questions were intended to establish a basic understanding of how each participant interacts with the computer on a day-to-day basis with the expectation that a relationship between computer use and animation preferences would become apparent.

The information consumption questions asked about frequency of Web browsing, examples of Web pages visited, and participation in other Internet activities such as

MUDs, Usenet news, and chat rooms. These questions were intended to examine the frequency with which the participants sought information with the expectation that a relationship between information consumption and animation preferences would become apparent.

After completing the questionnaire, participants were asked to use each display for an extended period of time in a variety of sizes, shapes, and configurations. The various types of available information and the ways for displaying it were outlined, and exploration was strongly encouraged. In monitoring information, participants were asked to consider how different types and configurations of animated displays are better for different types of informational situations. As noted previously, all user actions were timestamped and stored in a file in the participant's home directory. Before the files were generated and collected, participants were informed that the monitoring would take place and were given the opportunity to review the files before the data were processed.

In collecting this information, the hope was that the way in which participants interacted with the animated interfaces and the monitoring selections that they made would be related to their animation preferences. As a hypothetical example, perhaps participants who primarily use large displays also prefer the tickering animation, or participants who frequently interact with animations by clicking or dragging also prefer faster animations.

After having the opportunity to use tkwatch for several days, the users were interviewed during a 30 to 45-minute post-study session. Participants were asked a few general questions about their preferences for the devices, the factors that they felt contributed to their choice of display, and their motivations for monitoring different types of information. Primarily, however, the sessions gave participants the opportunity to expound upon the impacts and constraints of the animated displays, particularly on browsing habits, attention to other tasks, and general informational needs. The expectation was that comments from participants would help to understand their motivations for making the choices that they did. This in turn should simplify the processing of the data that were collected.

The next section profiles each user. Later sections look at some trends and categorizations that emerged based on participants behavior and consider how the knowledge gained might impact the design and use of peripheral animated displays.

7.3 User profiles

Ten students and two faculty members in the College of Computing participated in this study. Two of the participants were women, and three of the participants were born and raised outside of the United States. To help maintain anonymity, the names of the participants were all changed to generic male names that are common in America.

For the interested reader, this section provides a detailed profile for each of the participants in the observational study. Others can feel free to skip ahead to the summary and discussion sections following this one for a more general analysis of the results.

Milt Milt spent about 8 hours a day at the computer, most of it coding and debugging. He used an email biff, a clock with a second hand, and the grosview load monitor. He typically checked the Web at least four times a day, religiously in the morning, at lunch, and before he left, then whenever he had a free moment.

Milt used tkwatch for five days. He monitored sports scores, news headlines, and weather information. Although he generally got adequate information on topics from tkwatch, occasionally he was motivated to seek out a full story. Generally the information was interesting to know, but not things that he would have sought out on his own. Without tkwatch, Milt "would never have known that Boy George was almost killed by a glitter ball".

Milt liked both the fade and ticker, the fade because he could navigate quickly through the items by clicking on them and the ticker because he could see more information at once since multiple items can be on the screen simultaneously. Milt noted that news seemed better suited for ticker because he could just glance up and see one or more headers, while fade was better for sports scores where he could rapidly flip through the entire list. Milt did mention that the fade often caused "disappear anxiety" since items often disappear while he was reading them and, unlike with the ticker, there was no way to stop it.

Since Milt did not have a strong preference for fade or ticker, the factor that generally dictated which display type he used was space: he considered the size and shape of the available space and chose the animated display that best fit the space.

Milt used the automatic highlighting and history-based shadowing but did not notice them. He said he did not have much need to know when changes occurred.

Rod Rod spent about 8 hours a day at the computer, much of it coding and debugging with a good amount of Web browsing as well. Rod ran a clock, a grosview load monitor, and a MUD, which was constantly active and to which he contributed every few minutes. He checked the Web two or three times a day, generally whenever he had a free moment.

Rod used tkwatch for three days. He tried various types of information but in the end only monitored stock quotes. He greatly preferred the list display because he could see all of the information at once and because there was generally no motion on the screen. His second favorite was the ticker because he could grab, move, and stop it. He surmised that he might prefer a vertical scrolling (rolling) since that would be more like a MUD. The fade was his least favorite – he felt it gave him the least control over the information that he saw and showed him the least amount of information at any given time.

Rod liked the simplicity of the design and does not feel a need for additional visual cues like highlighting and shadowing. He did not want additional graphics that would take up space. He definitely did not want audio cues to indicate when information changed.

Stan Stan spent about 8 hours a day at the computer, most of it on email. He ran a large number of desktop tools, including an email biff, a clock, a load monitor, xpostit, the Irwin interface described in Section 3.1, and an Internet information delivery and discussion tool. He browsed the Web about once a day but did not visit any particular site on a daily basis and did not have any set times of day for browsing.

Stan used tkwatch for five days prior to the interview and at last count for over three weeks afterward. He monitored news, stocks, and weather information. Stan was very passive in his use of tkwatch, he would look at headlines but not feel the need to pursue the story elsewhere. Sometimes he would look at Irwin, which contains more information, to get more information, but did not go out to the Web or elsewhere.

In general, Stan preferred the ticker, with the fade second. The list was his least favorite because he felt it occupied too much of the screen. Going into the study, he thought that he would prefer the fade, but could not find a speed that was ideal for him. At first, he slowed it down from the default speed but found it distracting. He then sped it up and liked it better but was annoyed when the information would fade away before he had a chance to read it. Ideally, he would have liked to speed up the fading effect but lengthen the delay between fades. The ticker was a bit too fast at the default speed but was fine when he slowed it down. A big win with the ticker is that there is always something on the screen to read.

Stan found the automatic highlighting somewhat useful, but not the shadowing because it was too hard to read. He felt that other cues like images or colors might be useful. He envisioned color coding information based on type, with stocks yellow, sports scores blue, and so on. He did not see a need for more intrusive cues such as audio, in part because his other tools (in particular Irwin) already provided them.

Dan Dan spent about 11 hours per day at the computer, most of it coding and debugging. He used various desktop tools, including an email biff, a clock (with a second hand), and the grosview load monitor. Prior to using tkwatch, he ran the Yahoo stock ticker, but as tkwatch contains the same functionality, he stopped using it. He browsed the Web at least four times a day (whenever he had a free moment) in search of “news stories and high-tech gadgets”.

Dan used tkwatch for two days before the interview. He was particularly interested in monitoring stocks and the weather, but also monitored news for a while. He occasionally pursued news stories and other information that he saw in tkwatch, and he felt he would have done it more had it been easier to do and had he had more free time.

Dan liked the ticker best but wanted it to move faster, about at the speed at which he reads. He liked the ability to drag the ticker to move the information more quickly. He didn’t want to wait for the information to be able to read it. The fade he felt got in the way - “there were too many useless states” where the display was in an in-between state and nothing could be read. Ideally, he thinks he would prefer a fade with no intermediate states where the information would be blasted onto the screen with no gradual animation. He would also like the option of scrolling information (particularly stocks) vertically. Given some improvements, he feels he would use such a tool at all times.

Dan did not feel the automatic highlighting was necessary given the way he used tkwatch — he was not particularly interested in seeing an item as soon as it appeared but rather in browsing through the items when he had a free moment. The history-based shadowing was not advantageous because he could not read it with a single glance. He suggested color-coding information to emphasize categories. Just as Dan found that images were unnecessary in other peripheral displays, he did not like the idea of including images with the display.

Alan Alan spent 12 hours per day at a computer, most of it coding and debugging. He ran a biff, a clock, a load monitor, a scheduling tool, and an Internet information delivery and discussion tool. He browsed the Web at least four times per day.

Alan used tkwatch for five days. He monitored news, stock quotes, and the weather. He was pretty happy with the amount of knowledge that it provided and did not feel the need to seek out more information.

Alan's favorite display was the fade because he found it to be the least distracting, particularly at a slower speed. He felt that with the ticker he always had to wait for the information to appear and would lose focus. He noted "using the ticker required a higher cognitive load" and while the amount of information he processed and retained might be increased, to him it did not seem to be a fair tradeoff. He does not like tickers in other situations either, for example "the weather ticker comes on during a TV program, and if I look up for a moment I always miss something on the ticker." Alan did not like the list display: "it took up too much space." He is generally a passive user, but maybe once a week he would rapidly click or drag through the items.

Alan felt that his choice of display depended on the type of information he wanted to monitor. He felt that the ticker was better for information that is smaller and had a fixed size (stock quotes and sports scores) because there was less chance that part of it will be cut off. Also, he noted that people are familiar with tickers for information of that type, specifically stock tickers.

Alan turned on the highlighting and shadowing but seldom noticed them, probably because "there was not much new stuff". He did not find them distracting either, though. While he reported that he likes graphical alerts of changes like he had seen in other tools, he worried that additional graphical (and particularly audio) alerts in a tool like tkwatch would be distracting.

Fred Fred spent about 6 hours per day at the computer, most of it sending email and browsing the Web. The only other desktop tool he used is a clock, though he had used an Internet information delivery and discussion tool for a while. He browsed the Web several times a day, generally in the morning during breakfast then whenever he had a free moment.

Fred used tkwatch for three days. He monitored stock quotes, sports scores, news headlines, and the weather. He enjoyed pursuing more information on topics that he saw and wishes it was easier to do. He felt that tkwatch helped him to “surf smarter” by showing him when a ball game is over and a summary was likely to have been posted. Previously he would constantly pull up and reload a sports scores page to monitor the progress of games, a far more time-consuming activity. When Fred found the displays to be distracting, he would move them from the top of the screen to one of the lower corners.

Fred liked the fade display the best. He would enlarge the window and increase the font size to make it easier to read, slow it down to give him more time to read headlines, and regularly click through all of the information. He often sped up the display for sports scores because they change so often and because they are easy to read. Fred liked the ticker more than he thought he would. He appreciated the ability to adjust the speed and direction on the fly without using the configuration screen. He thought that the constant tickering motion would be a distraction, but after using it for a short time he found that it was not. The list was Fred’s least favorite. It took up too much room with larger fonts and was difficult to read with smaller ones.

Fred found that he used the ticker and fade differently. He looked at the ticker more often and quickly processed the multiple chunks of information on the screen at any given time. The fade forced him to look at and think about individual chunks.

Fred really liked the history-based shadowing for stock quotes. He was able to see see how his volatile Internet stock was changing over short periods of time. He tried the automatic highlighting but found that the italics was not enough of a clue. He reported that he would prefer other visual cues like color, and iconic flags for other information like new news. He also reported that he would like images associated with the information.

Tom Tom spent about 9 hours a day at the computer, most of it reading papers online and chatting with friends and family. He used an email biff, a clock, an ICQ/buddy tool, and an Internet information delivery and discussion tool. He browsed the Web less than once a day, generally just to get information about classes or to check on restaurants or fun activities.

Tom used tkwatch for four days. He most enjoyed the weather information but also monitors news, stocks, and sports scores. He felt that tkwatch provided him with interesting news that he would not otherwise see. He would have liked to be able to customize the presentation with filters to identify articles of particular interest.

The fade was Tom's favorite display. He preferred it at a speed slower than the default where he could look up occasionally and have time to read the entire contents of the display. The list confused him because it was difficult to determine when the information changed. He said the ticker was "horrid" because it moved too slowly and the display seemed crowded with multiple items in it.

While Tom did not like the ticker on his desktop, he acknowledged that it does have its place. He would enjoy it more in a public place like a cafeteria, but he found it required more sustained attention.

While Tom occasionally found the displays to be distracting, he did not turn them off, though he did cover them with other windows if need be. He felt that his primary tasks when sitting at the computer of reading papers and chatting with friends affected his preferences for a type of display.

Tom used the highlighting but did not notice it when it appeared. He found that history-based shadowing tended to clutter the display. He thought that a more subtle change showing age of the information may better differentiate old and new in situations like his where he did not need to know about changes as soon as they occur.

Jim Jim spent about 10 hours per day at the computer, with coding and debugging as his primary activity. He ran an email biff, clock, the grosview load monitor, and scheduling tool. He typically checked the Web over three times a day whenever he had a free moment.

Jim used tkwatch for four days. He monitored weather, news, and stocks for the company he used to work for and their biggest rival. He typically was happy with the level of knowledge that he obtained from tkwatch and did not feel the need to seek out more information.

Jim's favorite display was the fade. He liked it at a slow speed where he could read the information before it disappeared. The list was his second favorite, but he wanted to see more information than can be shown on the screen at once. He suggested that the fade and list be combined, where multiple units of information could appear on the screen, fading between different things. Jim thought the ticker was OK, but he would have preferred it to go faster. He felt the ticker was best suited for contiguous, related information like weather data and the forecast. His top priority was to see all information on a topic at once, which was why he liked the fade and list.

Jim felt that the shadow history was "a good concept, but was too jammed together" making it difficult to see the old information. He thought that the italics was more useful, but that color might be even better for highlighting changes. He did not see a need for auditory cues and felt that would be a distraction.

Matt Matt spent up to 4 hours per day at the computer, typically to do some specific activity that can only be done at the computer. As such, he did not clutter his screen with any displays not directly related to his current task – no email biff, no clock, no load monitor, nothing.

Matt used tkwatch for ten minutes. He did not like the constant motion of the ticker and fade, and the list occupied too much space on the screen. He liked the content in other similar interfaces slightly more, primarily because he could read information about local occurrences. To him, the ideal place for these types of displays would be in public places such as the picnic table study area of front entrance, where people could read them while looking away from their work or while waiting for friends.

Mack Mack spent about 6 hours per day at the computer, with major components being coding, writing, email, and Web surfing. He usually browsed the Web at lunch and whenever he had a direct need for it.

Mack used tkwatch for two days. He monitored news and weather, and wished it were easier to configure the tool to show specific information. He was skeptical about an automatic configuration feature, however. He did not feel a need to seek out more information than was on the screen. He had a work (and play) routine and he stuck to it.

Mack liked the fade best because it used the smallest region of the screen. He found the ticker to be distracting, but noted that he did not use it for very long and that it may get less distracting with longer use. He did not like the list because he could not tell when the information was changing.

Mack liked the automatic highlighting because it indicated which news items were newest. He felt that the history-based shadowing was not good for news and suggested associating it with certain types of information like stock quotes and sports scores. He would not like other highlights, particularly not audio ones. He turned off his email biff because the beeping annoyed him.

Mack used a virtual windowing system and chose to position tkwatch only on the screen that contains his email screen. He felt that with email, he had numerous pauses which gave him the opportunity to look at tkwatch. While working, he did not want the distraction, and while browsing the Web, he typically would be looking for specific things.

Will Will spent about 5 hours a day at the computer, most of it sending email. The only desktop accessory that he ran is a clock. He browsed the Web almost exclusively for research purposes and did not visit any sites on a daily basis.

Will used tkwatch for seven days. He primarily monitored news and weather, but occasionally would monitor a stock that he was interested in purchasing. He thought that he would enjoy the weather the most, but it did not change often enough. He monitored nine of the possible ten news topics (all of them except sports).

Will's favorite display was the ticker. He felt it was "not distracting, and provided more information" while the fade "pulled my eye to it more." He found the list the least useful because it was difficult to determine when a change occurred. "I assumed it would do something clever when something changed – it didn't!"

7.4 Summary of preferences

To facilitate the discovery of patterns and trends in the data, I summarized and compared the data generated by the users. This section presents the quantitative results that emerged from the pre-study questionnaire, the collected logfile data, and the post-study interview responses and discusses their impact. Table 9 tabulates the participant data. The table is divided into three portions: a ranking portion that shows the user-selected favorites from among the animated peripheral displays, a reported portion that shows reported computer usage, and a measured portion that shows usage statistics from the tkwatch log files. The following sections describe each portion.

7.4.1 Animation display rankings

Recall in the pilot study in Section 4.1, participants were asked to select their favorite peripheral display. For comparison, participants were again asked to select their favorite display, this time ranking them from one (favorite) to three (least favorite). Ties were allowed — they were allowed to select more than one display as their favorite. The first portion of Table 9 shows their selections.

Note that the fade display was the favorite of seven of the users, followed by the ticker with four and the list with one. This distribution is in stark contrast to the pilot study which resulted in fairly equal votes for each (fade was the favorite for seven participants, list six, ticker five). While the fade and ticker distributions are comparable, far fewer people preferred the list. There seem to be several differences between the two scenarios that may have contributed to this difference.

The first is the importance that people place on the different kinds of information. The tkscore interface was only used to show basketball scores during the NCAA Tournament, a high-interest event in which a single loss would eliminate a team. This seems to suggest that a user would want to see all pertinent information and changes as soon as possible, even with the additional space sacrifice that comes with the list display.

The second is the amount of information that can be displayed at any given time. In the tkscore study, the only available information was NCAA tournament basketball games, and there were at most eight games occurring at any given time. In tkwatch, the user could select literally hundreds of items considering the numerous types of news headlines and the almost limitless number of stock quotes. A list display can only show 10 or 12 items before it scrolls off the bottom of the screen. Granted, a user could have selected less information (as some did), but perhaps that is an unreasonable expectation in such a short-term study. I would expect that participants would gravitate toward the occasional monitoring of a small number of items at selected times.

7.4.2 Reported characteristics

The pre-study questionnaire was intended to establish a general understanding of the computer usage patterns of the participants. The “Reported” portion of Table 9 provides a summary of the responses to several of the primary questions. The complete survey tabulations can be found in Appendix C.

The first column of the “Reported” portion provides the number of hours per day that each participant spends at the computer. Since the participant pool was made up of computer scientists, the numbers are larger than one would expect from a typical cross-section of users. However, this allowed a large number of usage hours to be logged in a short time, speeding up the acclimation process. Fortunately, it was learned from the interviews that the participants differed significantly in their approaches to computing; that is, in what they did with their time at the computer.

The second column provides the participants’ estimates of the number of times they access the Web each day. Those that access it more than four times per day or less than once a day have a plus or minus by their entries. Note that the participants who spend a lot of hours at the computer generally spend a lot of time browsing the Web. However, there does not seem to be a trend connecting time at the computer or Web uses per day with a particular favorite display.

The third column indicates the number of peripheral desktop applications each participant typically runs. These applications can include email biffs, clocks, load monitors, chat tools, plan tools, and other similar applications that occupy a piece of the screen periphery.

7.4.3 Monitored characteristics

The monitored information includes the data collected from participant interaction with the animated peripheral displays. The "Monitored" portion of Table 9 provides a summary of some of the more interesting results gleaned from the data logs. A more detailed look at the monitored information can be found in Appendix C.

The first two columns in the "Monitored" portion of the table show the relative ticker and fade animation speeds that the participants typically selected by the end of the study. The expectation was that participants would experiment with various speeds before settling on a single desired speed. (The data and interviews seem to reflect this behavior.) The *normal* speed for fade and ticker is the default speed as identified by participants in the pilot study. Note that participants seem to prefer faster speeds, particularly for the ticker. Perhaps this is because there is much more information that is to be displayed, and participants are more anxious to see it all.

The third column indicates whether participants manually controlled the flow of information by clicking, grabbing, and dragging. A "yes" response indicates that participants controlled the flow more than five times, while a "no" response was less than or equal to five. The expectation was that participants may experiment with the manipulations on a few occasions, but would only do so repeatedly if they found it useful.

The fourth column reflects the median number of informational items monitored by a participant. An informational item is weather data, stock quotes, sports scores, or a news topic. As one might expect, participants who monitor a large number of items tend to dislike the list display, which requires screen space proportional to the

	Ranking			Reported			Monitored			
	Fade	List	Tick	Comp hrs	Web use	Perip apps	Tick speed	Fade speed	Manual control	Num items
Dan	2	3	1	11	4+	4	Fast	Fast	Yes	6
Jim	1	2	3	10	4	4	Fast	Norm	Yes	3
Rod	3	1	2	8	2	3	Norm	Norm	Yes	1
Mack	1	3	2	6	2	1	Fast	Fast	No	2
Stan	1	3	1	8	1	6	Slow	Slow	No	5
Tom	1	2	3	9	1-	4	Norm	Norm	No	9
Matt	-	-	-	4	1-	0	–	–	–	-
Will	2	3	1	5	1	1	Fast	Norm	No	9
Milt	1	3	1	8	4+	2	Fast	Norm	No	3
Alan	1	3	2	12	4+	4	Norm	Slow	No	5
Fred	1	3	2	6	4+	1	Norm	Norm	Yes	3

Table 9: The user-assigned rankings of favorite display types, reported computer usage statistics, and measured usage statistics of peripheral displays. A dash indicates a missing value. The plus (minus) next to some numbers indicates that the value is greater than (less than) or equal to the indicated value. The names are grouped according to the identified categories: the first group is information junkies, second greppers, third thinkers, and fourth computer-as-tool users. The last set of participants were not a close match for any of the groups.

number of items being monitored.

7.5 Categorizing users

Recall that one objective of this study is to be able to predict animation preferences using readily available characteristics about the users. To accomplish this, I grouped users according to their animation type and configuration preferences, then looked to see which personal characteristics they shared. After collapsing the categorizations and iterating the process, several interesting categories of users emerged. I attached names to the categories that are descriptive of the characteristics and behavior of their members. Information junkies, who spend a lot of time on the computer searching the Web, tend to prefer fast tickers. Information greppers, who are more focused in their computer time and in their monitoring and awareness, prefer the list display. Thinkers, who are less interactive in their time at the computer, prefer a fade with a long duration but short fade time. Computer-as-tool users, who use the computer only for select tasks, are most likely to suggest that animated displays be used in places other than the computer desktop. A more in-depth description and a summary profile of matching participants for each category follows:

- **Information junkies** use computers not only as tools but as sources of information and entertainment. They spend a lot of time (ten hours or more per day) in front of the computer and tend to have a long list of Web sites that they visit regularly. They seem to like the ticker but want it to go faster. They are likely to want a great deal of control over the information that appears in a display like tkwatch, even more than was provided. They are the ones who grab and pull and click through information to see it more quickly.

  Based on his behavior, Dan closely matches the information junkie profile. He spent eleven hours per day at the computer and browsed the Web more than four times per day. He liked using a fast ticker, and he was the most likely participant in the study to grab and move the ticker to read the information.

  Jim also is a close fit for the junkie category. He spent ten hours per day at the computer and browsed the Web four times per day. While he indicated a preference for the fade and list displays over the ticker, his problem with the ticker stemmed from the fact that it did not move fast enough.

- **Information greppers** monitor a small, very specific list of items. Named for the UNIX grep command used to search for patterns in large files [79], there are a few select pieces of information that greppers want to monitor, and they do not want to see other information. Information greppers tend to prefer the list display because it shows all of the information on the screen, making it easier to isolate items of interest. The space concerns that many users have typically will not affect greppers because they monitor so few items. Greppers are characterized by the fact that they keep few applications running at any given time and that the spend little time browsing the Web.

  Based on his behavior, Rod best exemplifies an information grepper. He focused on one task at a time and only browses the Web when he had a specific informational need. He was interested in seeing a select few stock quotes because he had money invested in these stocks and wants to know about changes as soon as possible. Mack also exhibited traits of an information grepper. He monitored a small list of items, though too many to show in the small space he was willing to dedicate to tkwatch. He had a preference for the fade over the list because of space concerns and because the motion drew his attention to it.

  The pilot study brought out the grepper role in many people. The importance of the tournament motivated people to keep close track on the information, and the small amount of information available in the display (only scores of a limited number of games were available) made the list a viable option. This may explain why so many more pilot study participants preferred the list compared to observational study participants.

- **Thinkers** prefer to obtain information in a hands-off manner. They tend to prefer the fade display running at slow speeds to give them time to read the information. They are not likely to interact with the display by clicking through the displayed information. They are characterized by their tendencies to spend long hours at or around the computer but to browse the Web infrequently. They tend to run a lot of peripheral information displays like load monitors, email biffs, and chat tools. The "thinkers" name comes because they do not necessarily interact with the computer constantly, even though they spend long hours near it. To them, the computer is more of an appliance that constantly presents information.

  Based on their behavior, Stan and Tom are both thinkers. Both mentioned that they slowed down the fade considerably to give themselves time to read the information. Neither liked to interact with the display by clicking on it, and neither seeks out more information about stories that they see in the display.

  Stan ran more peripheral desktop applications than any other user. He enjoyed being able to glance at various points of his computer screen to obtain information without having to explicitly seek it out. Stan suggested that he would like to "speed up the wipe but not the delay", that is, the information should be on the screen for a longer period of time but the fading action should be less pronounced.

  Tom also used a large number of peripheral desktop applications. Tom noted that he "didn't like to zip through the news" but instead preferred to glance at it occasionally to take in headlines at his leisure. His primary tasks at the computer were reading papers downloaded from the Web or emailed to him.

- **Computer-as-tool** people view computers as a tool to accomplish a specific task. Typically they spend little of their time in front of a computer, only the time that is necessary to accomplish the task at hand. They are unlikely to run many of the desktop applications that many people find useful: a clock, a biff, or a load monitor. Any value that they see in informational devices like tkwatch is separate from their daily tasks. In the study, these people were the ones who suggested presenting the information using non-desktop displays in locations where the information would be needed, such as traffic information near the parking lot or weather information near exit doors.

  Matt is a computer-as-tool person. He only sat at his computer to accomplish specific tasks. For him, the computer was "a tool, not part of the environment." He was not likely to linger there and did not want to drag out his current task with potentially distracting information. To him, the positioning of such information was important: he might enjoy "a giant screen next to the picnic tables" where people study and socialize.

  Will too is a computer-as-tool person. He spent much of his day away from the computer, and he did not occupy his screen with peripheral displays other than a clock. He reported that he is an avid user of mobile and handheld displays, and he seemed enthusiastic about non-desktop displays.

  Many people occasionally exhibit computer-as-tool person behavior. Consider that Mack, though he spent a good deal of time at the computer, said he would only use an application like tkwatch while doing a select set of tasks. He tried tkwatch on various screens of his virtual desktop but ended up leaving it on the screen containing his email because that was where he spent the most relaxing time. Similarly, Tom worked at several computers and only used tkwatch at the one where he read papers, chatted with friends, and replied to email. Both users used different computers for different tasks and view informational applications like tkwatch as appropriate and complimentary for only a select set of these tasks. Tom suggested that a tickering display would be ideal "in public places when I was alone or in a dull conversation" and would not mind focusing on such a display.

Table 9 groups the participants according to the category in which they best fit. Since not all of the participants were a good fit for a category, clearly there exist other categories of users. A closer, more in-depth characterization of the participants and a larger, more diverse user population would help to broaden and strengthen the set of categories.

In several cases, participants seemed to fall into more than one category. While this might seem problematic, it was to be expected. People who share a common preference for a particular type of animation are likely to share only a few characteristics. This actually seems to be advantageous to designers who are targeting potential users — an application designed for a particular population potentially will appeal to a much broader group. For example, the similarities between information junkies and thinkers suggests that an application designed for a junkie might also appeal to a thinker.

The categorization of an individual seems dependent on the situation. One would imagine that at some times in any given day or week, a person would be more like an information grepper, while at other times the same person would be more of a thinker. As was seen earlier, Tom stated that he only uses tkwatch when he was reading papers, chatting with friends, or sending email. At other times, Tom suggests that he would prefer to see animated displays away from his desktop. This suggests that Tom at times was a thinker, while other times he was more of a computer-as-tool individual.

7.6 Implications

Prior to this work, the selection of animated displays for peripheral awareness has been by trial and error. Consider the progression of the use of tickers and fades by Web sites such as Headline News and ESPN. They used tickers early on but switched to a fade where items stay on the screen for a fairly long time before rapidly fading to the next item. Web sites, including the Georgia Tech College of Computing site, include animated effects in their Web pages that are often removed soon thereafter, no doubt because viewers found it undesirable.

To provide direction to the inclusion of animation in peripheral displays, this study has sought to understand the relationships between personal characteristics and animation preferences that may prove helpful in constructing appropriate displays. To summarize the results of the study, several guidelines for the use of animation in peripheral displays can be suggested:

- **Fast highly-interactive tickers** should be employed if the target users spend a lot of time in front of the computer, frequently use the Web, and tend to use a lot of peripheral displays already.

- **Non-animated lists** should be employed if the target users spend an average amount of time at the computer, a below average amount of time browsing the Web, and only want to monitor a few select resources. If the list becomes too large, a fast fade can be substituted.

- **Fast fades with long delays** should be employed if the target users spend average to above-average time at or around the computer but are not always interacting with it.

- **Non-desktop delivery methods** should be employed for target user populations that do not spend all of their time at the computer and are not frequent Web users. While non-desktop displays were not tested, these people suggested that they would prefer non-desktop methods.

Using these guidelines, application builders can seek to understand their target user population and build the application to meet their needs. For example, in designing a tool for system administrators, one might expect them to fit the information junkie category, spending long hours at the computer and checking the Web frequently for breaking information. As such, one should perhaps design an application that uses a fast tickering effect to support peripheral awareness. By considering the characteristics of the target user population and designing applications that will be most suitable for them, designers can create useful and usable peripheral applications more quickly and easily.

In addition to focusing on more and better categorizations, further work should consider additional methods for supporting awareness. Several users mentioned that they did not like the list display because it was difficult to determine when information changed. Much of this problem may be the perception that a static display does not change while a dynamic one (the ticker and fade) does. After all, the ticker and fade may very well constantly cycle through the same information, but a user will be aware of the motion and will think that updates are taking place. To combat this problem, a static list display can incorporate auxiliary display indicators like flags to highlight new information or time data to indicate when the last change occurred. More important changes that demand immediate attention or that could affect the health and well-being of a viewer could even be accompanied by an audio cue, such as the warning beep that accompanies emergency weather bulletins on television.

CHAPTER 8
Conclusions and Future Work

The main focus of information communication research has been on the display of large information sources using innovative, often complex, visual displays. However, a communication mechanism need not be a rendered, 3-D encapsulation to be effective. In fact, in awareness situations such displays would be inappropriate. This thesis has examined the task of maintaining awareness of dynamic information sources, progressing from the early stages of examining potential communication mechanisms to the development of an understanding of the benefits of animation in maintaining awareness.

The focus of this thesis was on the use of small peripheral desktop displays that utilize motion-based and in-place animation techniques to convey the state of and changes to dynamic information resources. Experiments and user studies explored relative advantages of the different animation techniques and the impact of user characteristics on preferences and performance. A widget set supports programmers in building awareness applications by providing animation capabilities and related support.

The pilot study examined the use of animated displays as a way to maintain information awareness. The participants in the study seemed willing to sacrifice desktop space for heightened awareness. Most of the participants were content with the amount of information provided (though some desired more), and no one listed distraction caused by the animation as a reason for not using the program. The results described in this study were a stepping stone to better understanding the potential use of animation in maintaining awareness.

The empirical evaluations explored the balance between distraction, reaction, and comprehension and memorability when using peripheral animated displays. The first experiment showed that fading, tickering, and blasting peripheral displays can be non-distracting yet can effectively communicate information, and that the type of animation impacts performance on awareness tasks. The second experiment showed that changes in size and speed can impact performance on awareness tasks.

The development and analysis of Agentk provides programmers to include animated effects in their applications using a familiar programming interface. Use of the widgets makes it easier for end-users to have a sense of control over the information flow while maintaining a desired level of information awareness. The advantage gained from using Agentk is that constant, cyclic animations can be developed easily and rapidly, a definite advantage given the immediate needs that arise in the awareness problem domain.

The observational study provided a way to predict animation preferences using readily available characteristics about the users. Four user categories were identified that show promise in being able to associate user characteristics with animation preferences. This association would allow designers who have an understanding of their target population to construct displays that better communicate information.

Overall, this research has developed an understanding of the nature of the awareness problem and of how users' wants and needs differ in maintaining awareness. It is reasonable to conclude that the use of animation can assist in maintaining awareness without causing undue distraction in particular situations. While the experiments and studies described in this thesis looked only at short-term situations, numerous situations exist when a animated peripheral interface that is used even for short periods of time could potentially be beneficial: tracking breaking news stories on hot topics, monitoring the weather before a meeting in another building, or keeping an eye on volatile stocks. As with most tools, peripheral animations can be and have been misused, but when used properly, this thesis has shown they have the potential to be beneficial.

To increase the chances that peripheral animated displays will be used wisely and to improve the user perception of animated displays, programmers should not necessarily write applications that employ animation with the expectation that they will be used continually, but rather for short, well-defined periods of time. Perhaps they should be designed only to monitor the traffic between 5 pm and 6 pm every weekday, or to watch the scores of selected baseball games during the pennant drive and playoffs. If programmers do anticipate that it will be necessary to run the application at all times, alternate (non-animated) information delivery mechanisms should be made available as well.

Will people use peripheral animated displays to maintain awareness? The results of the experiments and studies suggest that they will, as do the popularity of certain peripheral displays that have been in use for many years. Consider instances when users are willing to tolerate constant animation. Glancing at a clock on the computer desktop to obtain the time is far easier than running a command or even looking at a wristwatch, and looking at a load monitor while running a compute-intensive program is far easier than running commands to determine the system load. Besides being immediately available, another distinction is that the changes to the display are small, subtle, and predictable, allowing the user to adapt to the changing display to the point where it is not in the least distracting. By providing smooth animations found in Agentk, the applications will be smoother and less likely to distract the user unnecessarily. The programmer and user can assist in this by choosing appropriate sizes and speeds for the application based on the anticipated informational goals of the user. It is necessary to understand the trade-off between the importance of the information being communicated and the resources necessary to display and process it. People will use applications and designers should create applications if they feel the trade-off is justified.

8.1 Extensions and improvements

The results in this thesis lay the groundwork for a good understanding of awareness and peripheral displays. There are some natural extensions to this work that would help expand and strengthen the results.

8.1.1 Empirical evaluations

The choice to study fading and tickering effects was based on the expectation that other in-place animations would show similar results to fade and other motion-based animations would show similar results to ticker. Initial evidence from these studies and elsewhere seem to support this claim; for example, the first experiment showed that the in-place blast effect resulted in similar performance to the in-place fade, and the Maglio work described in Chapter 2 showed that motion-based vertical tickering resulted in similar performance to horizontal [45]. However, further evaluations should be performed to explore the similarities and uncover possible differences.

Another area for further exploration is the effect of peripheral animation on various different central tasks. In my empirical evaluations, I selected browsing tasks to be performed because several pilot study participants indicated that for them the most likely time for using animated displays was while browsing the Web. However, many potential users may want and need to use animated displays at other times. While the evaluations in this thesis showed that browsing tasks are not affected by the presence of peripheral animations, the Maglio study showed that performance on another task, document editing, does seem to be affected [45]. Perhaps the most affected tasks are highly interactive tasks, or perhaps cognitively demanding tasks, or reading-intensive tasks. Further studies using different central tasks would help in understanding the degree to which different tasks are affected by the presence of animation.

Finally, while the experimental results in this thesis have indicated that performance differences exist based on characteristics of the animation being used, additional research is needed to explore *why* differences exist. Additional experiments that used eye tracking could determine how people budget their time between primary and peripheral tasks. Are certain animated effects or characteristics of animation more likely than others to attract an individual's attention? Do users read the information in peripheral displays or merely glance at them in search of certain key words or names? Are users less likely to glance at a peripheral display when performing a more cognitively demanding task? Or do they glance at the peripheral display at about the same rate but perform worse on the central task? Empirical evaluations similar to the ones described in this thesis, but augmented with eye tracking capabilities, could shed light on these and other similar questions.

8.1.2 Implementation support

The development of any user interface toolkit can be a never-ending process. There are always more features that can be added. This section discusses extensions and improvements to Agentk suggested by end-users and programmers.

The Agentk toolkit provides support for using animated effects in peripheral awareness applications. However, only a small subset of all possible behaviors are represented. Extending the widget set to include other behaviors such as shrinking, growing, start-and-stop, and swiping will provide a wider variety of techniques for programmers to choose from and should attract a larger population of programmers and end-users.

In addition, the collection of options should be extended to include Disney-style techniques such as motion trails and slow-in/slow-out [41]. These effects have the potential to lessen the distraction caused by the animations and improve the ability of the widgets to communicate by generating smoother changes.

Ultimately, I hope to create a framework within which any programmer can add new widgets. In creating Agentk, it has become clear that much of the code can be shared among widgets. Already I have isolated and encapsulated routines that have been and can be reused in other widgets. By only requiring programmers to specify a brief description of the behavior of the widget and the options, they are more likely to expand the behavior of the toolkit, benefiting not only themselves but future users.

8.2 Future work

Probing deeper, the results in this thesis also provide a strong foundation for future work in awareness and in peripheral displays. One area of future work is in combining the knowledge gained about animated peripheral devices with knowledge about the peripheral communication abilities of other media types. Another area is in applying the results studied here to the many real-world situations in which peripheral awareness is an important problem. This section discusses future work in these two areas.

8.2.1 Complementing other media types

While the use of peripheral animation has proven to be useful in helping to maintain awareness, future research should move beyond looking at animation alone and consider how it can be combined with other communication media to further enhance performance on awareness-related tasks and activities. The Agentk toolkit described in Chapter 6 included two graphical augmentations to the animated displays, automatic highlighting and history-based shadowing, that showed promise in informing the user of changes. Agentk also allowed text and images to be combined in animated widgets (as seen in the CWIC passive image browser described in Chapter 6), though the effectiveness of fading images has not been analyzed.

Auditory cues have proven useful in assisting with the monitoring of information [16]. As was seen in this thesis, there is a fine line between monitoring and awareness — often the tasks intertwine. As such, combining auditory cues with animated peripheral displays may help draw the user's attention in important situations. This was examined briefly in [45].

Ultimately, the goal is to automate the awareness process in the spirit of prior work in the automation of graphical information presentation by Mackinlay, Casner, and others [44, 12, 15, 71]. Much of the research to date has focused only on characteristics of the information and characteristics of the task at hand. This research extends the work by considering that users must accomplish not just a sole task but multiple tasks at any given time. I want to support easy identification of awareness mechanisms for arbitrary dynamic resources and arbitrary situations. As a first step, perhaps the user would specify only the size and position of an available display area and the information resource to be displayed, and an automated system would then identify the best communication mechanism for that situation. In the end, the process should be automated to the point that communication mechanisms are categorized based on various features of situations in which they would be used. The nature of the mechanisms that are selected for a given situation would depend not only on the information in the resource, but on the level of importance the user gives the information at the time and on general characteristics of the user. When information changes, it should be delivered to users in an appropriate manner befitting their current and future needs.

8.2.2 Applying results to situations

There are many possibilities for the application of this research to building systems to address awareness needs and related problems.

The user studies that were conducted took place in an academic setting, but the potential for use extends to other settings as well. Consider the implications of this work on a classroom lecture situation. Students could submit questions silently and electronically, and they would appear on the lecturer's display. When the question appeared, the lecturer could make the decision whether to address it immediately if it were a timely and relevant addition, or to wait until later if it were a point that was to be discussed in the near future.

In another example, consider the incorporation of peripheral animations in a shared break room. A wall-sized flat-panel display could present peripheral displays for the entire room, showing interesting headlines, images from the Web, local news of interest, and other topics that could spurn conversation and interactions. Using cameras and microphones, the wall-sized display could be connected with other displays in other common areas, helping to facilitate interaction and establish community.

As a third example, consider the application of animated peripheral displays in a virtual reality (VR) environment. One common use for VR environments is in virtual walkthroughs of places unfamiliar to the users to acclimate them. The results from this work suggest that certain types of animated peripheral displays would be better suited than others for informing and alerting the user of buildings, roads, and other elements in the environment.

Another area of interest for the application of the thesis results is in addressing the needs of different user populations. While the testing focused on a fairly narrow population, this work has potential benefits for different populations, such as older adults or users with special needs. The population that was examined in this thesis may be the ideal one for using animated displays as many of them grew up using highly interactive interfaces. Examining a group that does not have this background would provide an interesting point of comparison. Users with special needs, in particular users who have limited use of their hands or arms, may benefit from having peripheral animated displays. Information could be delivered to them without having to manually scroll. Many of the RSVP and automatic scrolling experiments described in Chapter 2 are targeted for this group [36, 28, 37].

8.3 Summary of contributions

This research has contributed to the field of human-computer interaction by exploring the role of animation in maintaining awareness of dynamic information. This is a relatively new area that has growing importance given the continual increase in available dynamic information. Since much of the work in maintaining awareness has been rather ad-hoc, this thesis helps to provide a framework for future investigations of the awareness properties of communication mechanisms as well as specific guidelines for the use of animation in peripheral displays.

In summary, the work described in this thesis made the following contributions:

- **Identified how characteristics of peripheral displays can affect performance on primary and peripheral tasks.** The empirical evaluations showed that the type and even the presence of animated peripheral displays did not negatively impact performance on a central browsing task. However, animation type did affect performance on peripheral tasks: the motion-based ticker resulted in better retention of information while the in-place fade resulted in quicker identification of changes. The size and speed of the animation also affected performance as smaller displays resulted in quicker identification and faster displays resulted in better retention.

- **Created a user interface toolkit that simplifies the inclusion of gradual, cyclic animated effects in peripheral awareness applications.** This freely-available toolkit contains several widgets that simplify the inclusion of constant, cyclic animated effects in user interfaces. The options and bindings of the widgets provide additional support that can be helpful in maintaining awareness.

- **Categorized users based on personal and situational qualities that then can be associated with animation preferences.** The expectation is that developers of peripheral animated displays will match their target user population with one of the categories and can then use the preferences of users in that category to guide their development choices.

These contributions extend the understanding of the utility of animated peripheral displays in assisting in information awareness situations.

Appendix A
Pilot Study Questionnaire

This appendix contains the data from the pilot study. Section A.1 contains the questionnaire emailed to the participants, and Section A.2 contains a tabulated list of responses to the questionnaire.

A.1 Questionnaire

The following is the post-study questionnaire emailed to the people who participated in the tkscore study.

```
Hi,

Thanks for using the tkscore basketball monitoring program last week.

As was mentioned in the agreement form, here are the followup questions
about how you used it and what other features you want.  This information
is important to our research and should only take a few minutes for you
to complete.  Please email your responses to me by Friday.

Feel free to refresh your memory of tkscore by running a demo version at:
  /net/hg116/tkscore/bin/tkscore-demo
```

Thank you for your input, and please continue using tkscore!

Scott

General questions:

How would you describe your interest in the NCAA Tournament?

- No real interest
- Casually pay attention to interesting tournament stories
- Want to know scores and stories on a daily basis
- Like to check on scores and stories several times a day
- Want to know about scores, game status, and more continually through the day

How often did you use tkscore?

- Once
- A few times
- Whenever certain games were on
- Whenever I logged in

Cumulatively, how much time did you spend using tkscore?

- 0-2 minutes
- 2-30 minutes

- 30 minutes - 2 hours
- More than 2 hours

How did tkscore meet your needs for tournament information?

- More information than I needed
- About the right amount of information
- Not enough information (please list additional desired information)

Display-specific questions:

Rate your preferences (1=most, 3=least favorite) of the three display types.

- Fade display
- List display
- Ticker display

What factors contributed to your choice of display? Choose all that apply.

- Ease with which information can be accessed and read
- Amount of information visible at any given time
- Level of distraction
- Control over visible information
- Control over rate of change

- Other (please list)

Future work questions (choose all that apply):

With what other information sources would you use tkscore-like displays?

- Stock quotes
- News headlines
- Weather information
- Summaries of email and newsgroup messages
- College of Computing announcements
- Activities of friends/colleagues
- MUD/MOO/Chat-room information
- Other (please list)

What other features of the information should be displayed?

- Highlights of selected information
- More emphasis on changes to information
- Summary of information state over time
- Other (please list)

```
What additional display mechanisms do you desire in a program like tkscore?

- Pictures and other graphics

- Auditory cues tied to important changes to information

- Easier to control speed of display updates

- More potential for user interaction (please list)

- Other (please list)



Other questions, comments, or suggestions:




Thanks again!

Scott
```

A.2 Questionnaire responses

The following is a tabulated list of responses to the questionnaire.

```
How would you describe your interest in the NCAA Tournament?

4 No real interest

5 Casually pay attention to interesting tournament stories

0 Want to know scores and stories on a daily basis

4 Like to check on scores and stories several times a day
```

8 Want to know about scores, game status, and more continually through the day

How often did you use tkscore?

6 Once

12 A few times

0 Whenever certain games were on

3 Whenever I logged in

Cumulatively, how much time did you spend using tkscore?

2 0-2 minutes

8 2-30 minutes

3 30 minutes - 2 hours

8 More than 2 hours

How did tkscore meet your needs for tournament information?

3 More information than I needed

10 About the right amount of information

7 Not enough information (please list additional desired information)

1 N/A

for those with no real interest:

2 More information than I needed

1 About the right amount of information

0 Not enough information (please list additional desired information)

1 N/A

for those who casually pay attention:

1 More information than I needed

4 About the right amount of information

0 Not enough information (please list additional desired information)

0 N/A

for those who like to check several times a day:

0 More information than I needed

2 About the right amount of information

2 Not enough information (please list additional desired information)

0 N/A

for those who want to know continually:

0 More information than I needed

3 About the right amount of information

5 Not enough information (please list additional desired information)

0 N/A

Rate your preferences (1=most, 3=least favorite) of the three display types.

most favorite votes:

7 Fade display

6 List display

5 Ticker display

3 N/A

What factors contributed to your choice of display? Choose all that apply.

15 Ease with which information can be accessed and read

12 Amount of information visible at any given time

9 Level of distraction

8 Control over visible information

4 Control over rate of change

4 Other (please list)

for those with fade #1 (max 7 for each)

6 Ease with which information can be accessed and read

2 Amount of information visible at any given time

5 Level of distraction

3 Control over visible information

2 Control over rate of change

3 Other (please list)

for those with list #1 (max 6 for each)

4 Ease with which information can be accessed and read

5 Amount of information visible at any given time

1 Level of distraction

2 Control over visible information

0 Control over rate of change

0 Other (please list)

for those with ticker #1 (max 5 for each)

4 Ease with which information can be accessed and read

4 Amount of information visible at any given time

2 Level of distraction

3 Control over visible information

1 Control over rate of change

1 Other (please list)

for those with no preference (max 3 for each)

1 Ease with which information can be accessed and read

1 Amount of information visible at any given time

1 Level of distraction

0 Control over visible information

1 Control over rate of change

0 Other (please list)

With what other information sources would you use tkscore-like displays?

9 Stock quotes

11 News headlines

12 Weather information

6 Summaries of email and newsgroup messages

8 College of Computing announcements

8 Activities of friends/colleagues

0 MUD/MOO/Chat-room information

5 Other (please list)

What other features of the information should be displayed?

5 Highlights of selected information

14 More emphasis on changes to information

8 Summary of information state over time

4 Other (please list)

What additional display mechanisms do you desire in a program like tkscore?

3 Pictures and other graphics

8 Auditory cues tied to important changes to information

6 Easier to control speed of display updates

4 More potential for user interaction (please list)

6 Other (please list)

for those with fade #1 (max 7 for each)

1 Pictures and other graphics

1 Auditory cues tied to important changes to information

1 Easier to control speed of display updates

0 More potential for user interaction (please list)

3 Other (please list)

for those with list #1 (max 6 for each)

2 Pictures and other graphics

3 Auditory cues tied to important changes to information

2 Easier to control speed of display updates

2 More potential for user interaction (please list)

2 Other (please list)

for those with ticker #1 (max 5 for each)

0 Pictures and other graphics

3 Auditory cues tied to important changes to information

2 Easier to control speed of display updates

1 More potential for user interaction (please list)

1 Other (please list)

for those with no preference (max 3 for each)

0 Pictures and other graphics

1 Auditory cues tied to important changes to information

1 Easier to control speed of display updates

1 More potential for user interaction (please list)

0 Other (please list)

Appendix B
Programming with Agentk

Agentk (pronounced agent-t-k) is extension to the Tcl/Tk toolkit that facilitate the incorporation of various graphical and motion-based effects into user interfaces. It requires Tcl/Tk 8.3 (or a later version) and requires Jeff Hobbs' megawidget creation package (included in the distribution) to operate. Agentk is available on the Web from the Agentk Web site:

`http://www.cc.gatech.edu/ mccricks/agentk/`

The remainder of this appendix describes the widgets, options, subcommands, and bindings for Agentk as well as several sample programs that are included with the distribution.

B.1 Agentk widgets

The fade widget fades in and out between several blocks of text and graphics. The gradual change will be less distracting than a sudden switch, yet will allow multiple information blocks to be displayed in a single area. The programmer controls the speed with which the fade occurs depending on the nature of the application: if the widget is used in a non-agent application interface, a quick fade might be used, while an agent might call for a slower, less intrusive fade.

The ticker widget provides a ticker tape display that scrolls or "tickers" information across the screen. As with the fade widget, a slow tickering can be less distracting than a sudden switch in information. The ticker widget is better suited for streams of text with arbitrary length – news headlines, weather reports, traffic updates. The programmer can control the rate of the tickering depending on the nature of the programming. The user can grab the ticker using the mouse to stop the tickering effect to make it easier to see the information. By dragging the information at a certain velocity, the user can change the speed and direction of the tickering effect.

Roll widgets are similar to ticker widgets, except they scroll information vertically. They work best for lists of information like golf leaderboards or top sales charts.

The navigation bar communicates information about the contents of a list by using the space in the trough of the scrollbar to represent the list entry. Each list entry is represented with a graphical line. Properties of the list entries are reflected in properties of the graphical lines.

B.2 Options, subcommands, and bindings

This section describes the subcommands, options, and bindings for the Agentk animated widgets.

A subcommand is a command that is available within a widget command. For example, all Tcl/Tk widgets (including the animated widgets) support the configure subcommand for querying and altering the widget options. Animated widgets additionally support the following subcommands:

- run command starts the animation effect for the widget.

- pause command pauses the animation until the run command is reissued.

- jump (fade only) jumps to the next item in the fade display list.

A configuration option alters the behavior and appearance of a widget. Common options include -geometry, -font, and -width. The following are some selected options that are available with the animated widgets.

- -speed indicates the speed with which the animation runs.
- -delay (fade only) is the delay in milliseconds before an item that has faded in begins to fade out.
- -text[variable], -bitmap[variable], and -image[variable] control the information that appears in the widget. When the value is changed, the new information animates into view as the old information disappears. For the variable options (such as -textvariable), the contents of each variable are animated on the screen. Only one of these options can be used for any given widget (the variable options have highest precedence, and images take precedence over bitmaps, and bitmaps over text).
- -separator contains a sequence of characters that separate entries in the ticker and roll widgets. This option is overridden by the -separatorbitmap and -separatorimage options, which can contain a bitmap or image separator.
- -markupstyle indicates a typeface markup (either bold, italic, or a color) used to highlight changes in information. Thus, if new text appears and the markupstyle is set to red, the new text will be red. To end the highlighting, -markupcount can be set to a number of iterations or -markuptimeout can be set to a length

Figure 21: The fade interface of the CWIC passive Web browser. The display consists of two fade widgets running in sync using the -drive option. The upper one fades between images and the lower one fades between text labels for the image. The first frame shows an initial image and text. The next two frames show how the image and text fade away as the new ones fade in. The final two frames show how the new image and text appear in their entirety as the old disappear.

of time after which the markups will disappear. Section 6.4.1 talks more about automatic markups.

- -shadowhistory (a boolean) indicates whether history-based shadowing should be enabled. If it is, previous states of the displayed information (text only) are shown using a shadow effect. See Figure 19 for an example. -shadowcount and -shadowtimeout can be used to remove the shadowing after some number of iterations or some length of time. Section 6.4.2 talks more about history-based shadowing.

- -throw (a boolean) indicates whether speed should be adjusted when the user "throws" a ticker or roll widget; that is, drags and releases it.

- -drive synchronizes widgets of the same type so that they will run in lock step. For example, the two fade widgets in Figure 21 must always run together as one widget shows images and the other shows labels for the images. As one drives the other, each step in the two fades will be calculated and displayed at the same time.

Bindings define the behavior when certain actions are performed by a user on a widget. Below are the default bindings. A programmer can tailor the bindings based on the needs of their applications.

- ButtonPress binding stops the animation effect. Users who want to read the text without being distracted by the animation can press and hold the mouse button.

- Motion of the mouse while the mouse button is being held down drags the ticker and roll widgets.

- ButtonRelease for the fade widget jumps to the next block of information. For the ticker and roll widget, it calculates a new speed based on the speed at which the widget was dragged. For all widgets it restarts the animation effect.

B.3 Demos

Agentk comes with several demo scripts that showcase its power. Most of them simply demonstrate the abilities of the widget, though some provide useful information monitoring capabilities. The Agentk demo programs are:

- **NewsAgent**: (Unix only) A simple news agent that monitors the Yahoo news site and constantly tickers the news headers across the screen. When new news arrives, a message rapidly fades in and out until a button is pressed. The button pulls up a simple news reader. Written by Emily Stretch.

- **checklpq**: (Unix only) Printer queue checker using a roll widget.

- **demo-fade**: Fade widgets that show images and captions in sync.
- **demo-navbar**: Navigation bar widget that highlights names in a list.
- **demo-roll**: Displays the script itself in a roll widget.
- **demo-ticker**: Displays the script itself in a ticker widget.
- **tkwatch**: (Unix only) Displays stock quotes, sports scores, news headlines, and more using either a fade, ticker, or list widget. This is the system used in the observational study described in Chapter 7.

Appendix C
Observational Study Data

This appendix contains the data from the observational study, including the questionnaire, a tabulation of the responses, and the data monitored using the tkwatch interface.

C.1 Observational study questionnaire

Below is the questionnaire that was emailed to participants in the observational study.

```
Thanks for volunteering!  Before getting started, I'd like to obtain
the answers to a few questions.  Many of these will be revisited in
the post-study interview, so think about them as you use the interface.


I. Computer usage (use recent and expected usage)

How many hours per day do you spend in front of a computer?


Estimate the percentage of time you spend doing various activities:
__% coding/debugging
```

__% writing (not code)

__% email

__% research on the Web

__% Web browsing (entertainment)

__% other (list major components)

What desktop information monitoring tools do you currently run?

[] email biff

[] clock

[] load monitor

[] ICQ/buddy tool

[] plan/scheduling tool

[] other (please list)

How frequently do you "act" on the information? That is, how often does the information you see result in you doing some other activity like checking your mail, killing a process, starting/contributing to a chat session, or going to a meeting?

[] once a day or less

[] 2-4 times a day

[] 1-2 times an hour

[] every few minutes

Do these type of tools ever distract you to the point that you turn them off? What aspects (sound, motion, etc) are distracting?

II. Information consumption

How often do you browse the Web?

[] less than once a day

[] daily

[] 2-3 times a day

[] 4+ times a day

What Web pages do you visit regularly (at least once a day)?

At what times do you browse the Web (select all that apply)?

[] when I first log in

[] at/around lunch

[] before I log out for the day

[] other set times (list examples)

```
[] whenever I have a free moment
```

```
What other Internet activities do you participate in regularly (daily)?

[] MUDs/MOOs
[] Usenet news
[] online games
[] chat rooms/groups
[] other (please list)
```

C.2 Questionnaire responses

Table 10 tabulates the factors that users indicated in the questionnaire contributed to their choice of display.

The first column reflects the impact of ease of readability on the choice of a display. Almost everyone rated this highly regardless of the type of animated display that they prefer. This suggests that each individual has a different opinion of the readability of displays.

The second column reflects whether the amount of visible information on the screen at any given time was a factor in choosing a display. As one might expect, the participants who value this tend to prefer the list or ticker displays, in which multiple items can be displayed.

The third column indicates whether the distraction caused by the animation was

	Read ease	Amount vis info	Distract level	Control vis	Control rate	Display size	Display shape	CPU load
Milt	2	1	0	0	0	1	2	0
Rod	2	2	2	2	0	2	0	0
Stan	1	2	2	0	2	2	2	1
Dan	2	1	1	1	1	0	0	2
Alan	2	0	2	0	1	2	1	1
Fred	1	1	0	2	2	2	1	0
Tom	2	1	0	0	0	1	1	1
Jim	2	2	1	1	2	1	0	0
Mack	1	1	2	0	1	2	1	0
Will	1	2	2	1	1	1	2	0
Totals	16	13	12	7	10	14	10	5

Table 10: Factors that users said contributed to their choice of display. A 2 indicates that the factor contributed greatly, a 1 indicates some contribution, and a 0 indicates little or no contribution.

a factor in choosing a display. It was expected that the constant motion of the ticker would be distracting, and only participants who were less sensitive to the potential distraction of an animation would prefer ticker. However, this was not the case: participants who gave this factor a high rating included participants who favored ticker, fade, and list.

The fourth column indicates whether control over the amount of visible information was a factor in choosing a display. This was rated the least important factor, perhaps because participants did not want to expand the size of the display.

The fifth column indicates whether control over the rate of change of the animation was a factor in choosing a display. Note that this is different from the desire to interact with the display as it is running. Control over the rate of change refers to the speed at which the animation moves, while the desire to interact with the display refers to the ability to click and grab the display to change it. Results from two factors are not correlated.

The sixth and seventh columns indicate the participant rating of display size and shape on display choice. Most participants indicated that they cared about this primarily with regard to the size and shape of the list, which dominated the display if more than a few items were being monitored. However, occasionally participants who did not have a strong preference to ticker of fade indicated that they simply chose the one that best fit onto their screen.

The eighth column indicates the importance rating for CPU load. The ticker can require a significant amount of computational power in updating the display, and participants in the pilot study had mentioned it as a factor for not choosing the ticker. However, in this study only one participant indicated that it was a highly important factor, and he stated in his interview session that this was because he wanted the ticker to run more quickly than it could.

C.3 Monitored characteristics

The monitored information examines the ways in which the participants used (or did not use) the animated peripheral displays. As a reminder, Table 11 lists the favorite displays selected by each user. Table 12 provides a summary of the responses to several of the primary questions.

The first two columns show the animation speed selected by the participant late in the study. The expectation was that participants would experiment with various speeds before settling on a single desired speed. (The data and interviews seem to

	Fade	List	Ticker
Milt	1	3	1
Rod	3	1	2
Stan	1	3	1
Dan	2	3	1
Alan	1	3	2
Fred	1	3	2
Tom	1	2	3
Jim	1	2	3
Mack	1	3	2
Will	2	3	1
Avg	1.4	2.6	1.8

Table 11: User favorites for each of the display types. Occasionally users listed more than one favorite, in which case both have the same value.

	Ticker speed	Fade speed	Ticker size	Manual control	tkwatch items	Use days
Milt	Fast	Normal	Normal	No	3	5
Rod	Normal	Normal	Normal	Yes	1	3
Stan	Slow	Slow	Small	No	5	5
Dan	Fast	Fast	Normal	Yes	6	3
Alan	Normal	Slow	Normal	No	5	5
Fred	Normal	Normal	Large	Yes	3	3
Tom	Normal	Slow	Normal	No	9	4
Jim	Fast	Slow	Normal	Yes	3	4
Mack	Fast	Fast	Small	No	2	2
Will	Fast	Normal	Normal	No	9	7

Table 12: System usage characteristics recorded from the participants' recorded usage sessions. Note that they may occasionally differ from the reported favorites. In this table "normal" speeds and sizes refer to the default values, with all other measures relative to the default. As one user (Matt) did not use tkwatch for more than a few minutes, insufficient data were available to include him in the table.

reflect this.) The *normal* speed for fade and ticker is the default speed as identified by participants in the pilot study. Note that participants seem to prefer faster speeds, particularly for the ticker. Perhaps this is because there is much more information that is to be displayed, and participants are more anxious to see it all.

The third column shows the typical size for the ticker animation. The normal size 80 characters, and participants only seemed to resize it to fit in in available spaces, not for readability or distraction reasons. (Since the fade display always shows a single block of information regardless of its size, it did not make sense to resize it and thus fade size is not in the table.) In general, participants seemed content with the size of the ticker. In the cases when they were not, they indicated a desire to fit the display to a particular area of the screen, not a desire to make it more comprehensible or less distracting as one might expect.

The fourth column indicates whether participants manually controlled the flow of information by clicking, grabbing, and dragging. A "yes" response indicates that participants controlled the flow more than five times, while a "no" response was less than or equal to five. The expectation was that participants may experiment with the manipulations on a few occasions, but would only do so repeatedly if they found it useful.

The fifth column reflects the median number of informational items monitored by a participant. An informational item is weather data, stock quotes, sports scores, or a news topic. As one might expect, participants who monitor a large number of items tend to dislike the list display, which requires screen space proportional to the number of items being monitored.

The sixth column indicates the number of days that the participant used tkwatch.

This value was reported by the participants and confirmed by the collected data. I wanted to be sure that each participant had used tkwatch on multiple occasions over an extended period of time.

Bibliography

[1] After Dark Online. Available at www.afterdark.com.

[2] BackWeb. Available at www.backweb.com.

[3] Ronald M. Baecker and Ian Small. Animation at the interface. In Brenda Laurel, editor, *The Art of Human-Computer Interface Design*, pages 251–267. Addison-Wesley, 1990.

[4] Ronald M. Baecker, Ian Small, and Richard Mander. Bringing icons to life. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '91)*, pages 1–6, New Orleans, LA, May 1991.

[5] Lyn Bartram. Enhancing visualizations with motion. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '98)*, pages 13–16, Raleigh, NC, 1998.

[6] Richard C. Bodner and I. Scott MacKenzie. Using animated icons to present complex tasks. In *Proceedings of CASCON '97*, pages 281–291, Toronto, CA, 1997.

[7] Steven A. Brewster. *Providing a structured method for integrating non-speech audio into human-computer interfaces.* PhD thesis, University of York, UK, 1994.

[8] Megan L. Brown, Sandra L. Newsome, and Ephraim P. Glinert. An experiment into the use of auditory cues to reduce visual workload. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '89)*, pages 339–346, Austin, TX, May 1989.

[9] Quasedra Y. Brown and D. Scott McCrickard. Cwic: Continuous web image collector. In *In Proceedings of the ACM Southeast Conference (ACMSE 2000)*, pages 244–252, Clemson, SC, April 2000.

[10] Michael D. Byrne, Richard Catrambone, and John T. Stasko. Do algorithm animations aid learning? Technical Report 96-18, Georgia Tech GVU Center, Atlanta, GA, 1996.

[11] Jonathan J. Cadiz, Susan R. Fussell, Robert E. Kraut, F. Javier Lerch, and William L. Scherlis. The awareness monitor: A coordination tool for asynchronous, distributed work teams. Available from www.cs.cmu.edu/ kraut, 2000.

[12] Stephen M. Casner. Task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics*, 10(2):111–151, April 1991.

[13] Bay-Wei Chang, Jock D. Mackinlay, Polle T. Zellweger, and Takeo Igarashi. A negotiation architecture for fluid documents. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '98)*, San Francisco, CA, November 1998.

[14] Bay-Wei Chang and David Ungar. Animation: From cartoons to the user interface. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '93)*, pages 45–55, Atlanta, GA, November 1993.

[15] Mei C. Chuah and Steven F. Roth. On the semantics of interactive visualizations. In *Proceedings of the 1996 IEEE Symposium on Information Visualization*, pages 29–36, San Francisco, CA, October 1996.

[16] Jonathan Cohen. Monitoring background activities. In Gregory Kramer, editor, *Auditory Display: Sonification, Audification, and Auditory Interfaces*, volume XVIII, pages 499–531. Addison-Wesley, 1994.

[17] Fred Douglis, Thomas Ball, Yih-Farn Chen, and Eleftherios Koutsofios. The AT&T internet difference engine: Tracking and viewing changes on the web. *World Wide Web*, 1(1):27–44, January 1998.

[18] Thomas Driemeyer. plan man page, September 1993.

[19] Robert L. Duchnicky and Paul A. Kolers. Readability of text scrolled on visual display terminals as a function of window size. *Human Factors*, 25(6):683–692, 1983.

[20] Stephen G. Eick, Joseph L. Steffen, and Eric E. Sumner Jr. SeeSoft—a tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering*, 18(11):957–968, November 1992.

[21] R. S. Fish, R. E. Kraut, and B. L. Chalfonte. The videowindow system in informal communications. In *Proceedings of the ACM Conference on Computer Supported Collaborative Work (CSCW 90)*, pages 1–11, 1992.

[22] Geraldine Fitzpatrick, Sara Parsowith, Bill Segall, and Simon Kaplan. Tickertape: Awareness in a single line. In *Conference Companion of the ACM Conference on Human Factors in Computing Systems (SIGCHI '98)*, pages 281–282, Los Angeles, CA, April 1998.

[23] K. I. Foster. Visual perception of rapidly presented word sequences of varying complexity. *Perception and Psychophysics*, 8:215–221, 1970.

[24] William B. Frakes and Ricardo Baeza-Yates. *Information Retrieval Data Structures and Algorithms.* Prentice Hall, Englewood Cliffs, NJ, 1992.

[25] Jim Fulton. xbiff man page, September 1988.

[26] George W. Furnas. Generalized fisheye views. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '86)*, pages 16–23, Boston, MA, April 1986.

[27] Cleotilde Gonzalez. Does animation in user interfaces improve decision making? In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '96)*, pages 27–34, Vancouver, BC, April 1996.

[28] Michael M. Granaas, Timothy D. McKay, R. Darrell Laham, Lance D. Hurt, and James F. Juola. Reading moving text on a CRT screen. *Human Factors*, 26(1):97–104, 1984.

[29] Anthony J. Hayter. *Probability and Statistics for Engineers and Scientists.* PWS Publishing Company, Boston, MA, 1996.

[30] Jeremy M. Heiner, Scott E. Hudson, and Kenichiro Tanaka. The information percolator: Ambient information display in a decorative object. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '99)*, pages 141–148, Asheville, NC, November 1999.

[31] Jonathan Helfman. Montage, February 1996. Presented at the 1996 Human Computer Interaction Consortium, available from portal.research.bell-labs.com/orgs/ssr/people/jon/.

[32] Jonathan Helfman. Mandala: An architecture for using images to access and organize web information. In *Proceedings of the 1999 International Conference on Visual Information Systems (VISUAL 99)*, June 1999.

[33] William C. Hill and James D. Hollan. History-enriched digital objects: Prototypes and policy issues. *The Information Society*, 10(2), April 1994.

[34] Scott E. Hudson and John T. Stasko. Animation support in a user interface toolkit: Flexible, robust, and reusable abstractions. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '93)*, pages 57–67, Atlanta, GA, November 1993.

[35] Hiroshi Ishii and Brygg Ulmer. Tangible bits: Towards seamless interfaces between people, bits, and atoms. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97)*, pages 234–241, Atlanta, GA, March 1997.

[36] James F. Juola, Nicklas J. Ward, and Timothy McNamara. Visual search and reading of rapid serial presentations of letter strings, words, and text. *Journal of Experimental Psychology: General*, 111(2):208–227, 1982.

[37] T. Jin Kang and Paul Muter. Reading dynamically displayed text. *Behaviour and Information Technology*, 8(1):33–42, 1989.

[38] Colleen M. Kehoe, John T. Stasko, and Ashley G. Taylor. Rethinking the evaluation of algorithm animations as learning aids: An observational study. Technical Report 99-10, Georgia Tech GVU Center, Atlanta, GA, 1999.

[39] Andruid Kerne. Interface ecology: Collage machine expands media browsing with temporality and indeterminacy. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97)*, Atlanta, GA, April 1997.

[40] Ioi K. Lam. Designing mega widgets in the Tix library. In *Proceedings of the 3rd USENIX Tcl/Tk Workshop*, Toronto, CA, July 1995.

[41] John Lasseter. Principles of traditional animation applied to 3d computer animation. *ACM Journal of Computer Graphics*, 21(4):35–43, July 1987.

[42] Andrea W. Lawrence, Albert M. Badre, and John T. Stasko. Empirically evaluating the use of animations to teach algorithms. In *Proceedings of the IEEE Symposium on Visual Languages (VL '94)*, pages 45–54, St. Louis, MO, October 1994.

[43] Henry Lieberman. Autonomous interface agents. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97)*, pages 67–74, Atlanta, GA, March 1997.

[44] Jock D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, April 1986.

[45] Paul P. Maglio and Christopher S. Campbell. Tradeoffs in displaying peripheral information. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2000)*, April 2000.

[46] D. Scott McCrickard. Information awareness on the desktop: A case study. Technical Report 95-24, Georgia Tech GVU Center, Atlanta, GA, 1997.

[47] D. Scott McCrickard. An extensible interactor toolkit for enhancing information awareness. Presented at the 1998 Human Computer Interaction Consortium (HCIC '98), 1998. Available as Georgia Tech GVU Tech Report GIT-GVU-98-28.

[48] D. Scott McCrickard. Agentk: A toolkit for enhancing agent interfaces. *;login: The Magazine of USENIX and SAGE*, 25(4), July 2000.

[49] D. Scott McCrickard and Gregory D. Abowd. Assessing the impact of changes at the architectural level: A case study on graphical debuggers. In *Proceedings of the International Conference on Software Maintenance (ICSM 96)*, Monterey, CA, November 1996.

[50] D. Scott McCrickard and Richard Catrambone. Beyond the scrollbar: An evolution and evaluation of alternative navigation techniques. In *Proceedings of the IEEE Symposium on Visual Languages (VL '99)*, pages 270–277, Tokyo, Japan, September 1999.

[51] D. Scott McCrickard and Thomas H. Rowan. Monitoring and visualizing information resources. Technical Report ORNL/TM-13193, Oak Ridge National Laboratory, July 1996.

[52] D. Scott McCrickard, John T. Stasko, and Q. Alex Zhao. Exploring animation as a presentation technique for dynamic information sources. Technical Report 99-47, Georgia Tech GVU Center, Atlanta, GA, 1999.

[53] D. Scott McCrickard and Q. Alex Zhao. Supporting information awareness using animated widgets. In *Proceedings of the 2000 USENIX Conference on Tcl/Tk (Tcl2K)*, pages 117–127, Austin, TX, February 2000.

[54] Brad A. Myers. The importance of percent-done progress indicators for computer-human interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '85)*, pages 11–17, April 1985.

[55] Brad A. Myers, Robert C. Miller, Rich McDaniel, and Alan Ferrency. Easily adding animations to interfaces using constraints. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '96)*, Seattle, WA, November 1996.

[56] Elizabeth D. Mynatt, Maribeth Back, Roy Want, Michael Baer, and Jason B. Ellis. Designing audio aura. In *Proceedings of the 1998 Conference on Human Factors in Computing Systems (CHI '98)*, pages 566–573, April 1998.

[57] Bonnie A. Nardi, J. R. Miller, and D. J. Wright. Collaborative, programmable intelligent agents. *Communications of the ACM*, 41(3), March 1998.

[58] John K. Ousterhout. *Tcl and the Tk Toolkit.* Addison-Wesley, Reading, MA, 1994.

[59] John K. Ousterhout. Scripting: Higher level programming for the 21st century. *IEEE Computer*, March 1998.

[60] Susan Palmiter. The effectiveness of animated demonstrations for computer-based tasks: A summary, model and future research. *Journal of Visual Languages and Computing*, 4:71–89, 1993.

[61] Susan Palmiter and Jay Elkerton. An evaluation of animated demonstrations for learning computer-based tasks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '91)*, pages 257–263, New Orleans, LA, May 1991.

[62] Sara Parsowith, Geraldine Fitzpatrick, Bill Segall, and Simon Kaplan. Tickertape: Notification and communication in a single line. In *Asia Pacific Computer Human Interaction 1998 (APCHI '98)*, Kangawa, Japan, July 1998.

[63] Pointcast network. Available at www.pointcast.com.

[64] Keith Rayner. Eye movements in reading and information processing. *Psychological Bulletin*, 85:618–660, 1978.

[65] A. Repenning. *Agentsheets: A tool for building domain-oriented dynamic visual environments.* PhD thesis, University of Colorado Department of Computer Science, Boulder CO, 1993.

[66] George G. Robertson, Stuart K. Card, and Jock D. Mackinlay. Information visualization using 3d interactive animation. *Communications of the ACM*, 36(4):57–71, April 1993.

[67] A. Sekey and J. Tietz. Text display by saccadic scrolling. *Visible Language*, 16:62–76, 1982.

[68] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating 'word of mouth'. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 95)*, Denver, CO, May 1995. ACM Press.

[69] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Addison-Wesley, Reading, MA, 1998.

[70] D. C. Smith, E. F. Harslem, C. H. Irby, and R. B. Kimball. The star user interface: an overview. In *Proceedings of the 1982 National Computer Conference*, pages 515–528, 1982.

[71] Bob Spence, Lisa Tweedie, Huw Dawkes, and Hua Su. Visualization for functional design. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 4–10, Atlanta, GA, October 1995.

[72] John T. Stasko. The path-transition paradigm: A practical methodology for adding animation to program interfaces. *Journal of Visual Languages and Computing*, 1:213–236, 1990.

[73] John T. Stasko. Animation in user interfaces: Principles and techniques. In Len Bass and Prasun Dewan, editors, *User Interface Software*, number 1 in Trends in Software, chapter 5, pages 81–101. John Wiley, 1993.

[74] John T. Stasko, Albert Badre, and Clayton Lewis. Do algorithm animations assist learning? an empirical study and analysis. In *Proceedings of the Conference on Human Factors in Computing Systems (INTERCHI '93)*, pages 61–66, Amsterdam, Netherlands, April 1993.

[75] John T. Stasko and D. Scott McCrickard. Real clock time animation support for developing software visualizations. *Australian Computer Journal*, 27(3):118–128, November 1995.

[76] Ivan E. Sutherland. Sketchpad: A man-machine graphical communication system. In *AFIPS Spring Joint Computer Conference*, pages 329–346, 1963.

[77] Edward R. Tufte. *Envisioning Information*. Graphics Press, Cheshire, CT, 1990.

[78] UNIX System V. biff man page, September 1987.

[79] UNIX System V. grep man page, October 1987.

[80] UNIX System V. wall man page, September 1987.

[81] Yahoo. Available at yahoo.com.

[82] Q. Alex Zhao and John T. Stasko. What's happening? the community awareness application. In *Conference Companion of the ACM Conference on Human Factors in Computing Systems (SIGCHI 2000)*, pages 253–254, April 2000.

Vita

Scott McCrickard was born in Jacksonville, Florida, but he quickly moved to Chapel Hill, North Carolina to correct the situation. For most of his early years, Scott lived in Greensboro, North Carolina, until he graduated from Grimsley High School in 1988. He returned to Chapel Hill, where he received a Bachelor of Science degree in 1992 from the University of North Carolina as a mathematical sciences major with an emphasis in computer science. Scott came to Atlanta to attend Georgia Tech, where he completed his Masters degree in Computer Science in December 1995 and will receive his his Doctorate in Philosophy in Computer Science in August 2000. Scott has accepted an assistant professor position at Virginia Tech in Blacksburg, Virginia, starting in Fall 2000.