Towards the Development of User Interface Design Guidelines for Large Shared Displays

Khaled Hussein

Thesis Submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Computer Science and Applications

D. Scott McCrickard, Chair
Christopher L. North, Member
Ayman A. Abdel-Hamid, Member

June 5, 2007
Blacksburg, Virginia

Keywords: Large Shared Displays, Collaborative Environments, Personal Displays, Interface Design Guidelines

Towards the Development of User Interface Design Guidelines for Large Shared Displays

Khaled Hussein

ABSTRACT

As large displays become more affordable, researchers investigate their productivity impacts and try to develop techniques for making the large display user experience more effective. Studies show that large displays enable users to create and manage more windows and engage in more complex multitasking behavior. Although recent work demonstrates significant productivity benefits of large shared displays, it shows numerous usability issues because current software design is not scaling well. Therefore, we took steps towards developing two user interface design guidelines for large shared displays.

Specifically, empirical studies have been conducted to compare the effects of large shared display and personal display use. When each of them is used as a secondary display, large shared displays impose increased interruption and comprehension. Empirical and qualitative studies are designed to develop two user interface design guidelines for large shared displays. We designed a system called SuperTrack that uses a large shared display and the proposed guidelines to further enhance team efficiency and productivity in collaborative software development environments. Finally, an in-situ evaluation assesses the benefits of SuperTrack. Results show that exposing software development team members to a large shared display through SuperTrack leads to more communication among the members and improved group awareness – leading to increased productivity and efficiency.

DEDICATION

To GOD,

By whom everything is possible

To my parents,

For supporting me all the way

To my family,

For providing more meaning to my life

To my friends,

For always standing beside me

ACKNOWLEDGMENT

I am who I am in large part because of the many wonderful people in my life. Thanks GOD for putting these people in my life. I have a number of individuals to thank for helping me to achieve this goal and for making my time at VT-MENA so memorable.

First and foremost, to my parents … you have stood behind me from day one, always encouraging me to take the next step and follow my dreams. You allowed me to pursue my own path, supported me through all my bad days, and commended me when I finally succeeded. I love you; I thank you and I hope I've made you proud.

I would like to thank Dr. McCrickard, my advisor, for his incredible help and guidance as well as for his constant availability and attention to quality work. Thank you for making this entire research effort such a challenging, shaping, and influential yet enjoyable experience.

It is a great pleasure to also acknowledge Shahtab Wahid for his valuable advice and support. Thank you for the good times and your constructive comments on my work.

My appreciation also goes to my committee members, Dr. Abdel Hamid and Dr. North for their invaluable advice, comments, and guidance on my research work.

I would like to thank Dr. Hanafy and Dr. Sedki for making it possible for me to attend the enjoyable graduate studies in Virginia Tech while still being in Egypt. My thanks go to all my friends in VT-MENA for making my time in VT-MENA really productive and enjoyable.

Finally, I would like to thank all my friends that helped me to pass through tough days. Thanks you for your love and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

Introduction

This chapter presents the objective of our work. We address the problem of understanding different psychological effects when using different types of secondary displays. We then talk about the need to solve this problem and the motivation behind our research. Moreover, this chapter elaborates on the thesis statement along with the research approach followed. It also sheds light on the contributions and future impacts of our work to the Human-Computer Interaction research. Finally, it provides an organization of the remaining structure of this thesis.

1.1 Problem Description

Human-Computer Interaction researchers have always tried to find ways to enhance user productivity and efficiency when interacting with computers. The introduction of using multiple displays has created a new set of questions and attracted the interest of many HCI researchers. Grudin started to investigate answers to some questions about how the users arrange and use the secondary displays (Grudin 2001). Results show that users tend to use secondary displays for secondary tasks that do not require primary attention.

Although, by that time, there were some shortcomings in measuring the benefits of using multiple monitors, Grudin believed that there were substantial subjective benefits.

Given that users may gain different types of benefits from using multiple displays, application designers faced an interesting new problem. The existing applications were not designed with the use of multiple monitors in mind. Therefore, the interfaces need to be redesigned considering the use of multiple monitors. However, there are not enough guidelines for interface designers to follow to meet the challenge. Hence, HCI researchers were asked to investigate and develop user interface design guidelines for secondary displays.

Although there are not enough guidelines for secondary-display interface designers, the advancements in display technologies make the situation even more challenging. More questions started to arise. For example, should the guidelines that work on a type of secondary display apply to all other types? What is the best type of secondary display for specific situations? What are the effects on the user when using different types of secondary displays? Therefore, we need to understand the different types of existing guidelines and how we can use and develop them further to encompass different types of displays.

More importantly, we see an urgent need to understand the differences between the various types of displays. With this understanding, we begin to evaluate different types of displays in different environments. We believe that if we can understand what type of display is appropriate for each situation, we can lessen the gap between the rapid technology advancements and the lack of research that maximizes the benefits of this technology.

1.2 Motivation

There is a trend towards expanding the use of multiple displays. There is much research in the literature that started documenting different types of user benefits from the use of multiple displays. A more detailed discussion of these benefits is presented in the

Related Work chapter. This research served as a stimulus to motivate users to use multiple displays on the work level rather than simply a personal level. In turn, corporations started to use different types of displays for different purposes in the work environment. For example, large secondary displays have been adapted to work as notification systems (Greenberg and Rounding 2001).

Research in a variety of areas looked into the benefits of using different types of multiple displays in multiple situations. For example, software engineering researchers looked at the benefits of using shared displays in software development environments and using secondary personal displays in pair-programming environments. The promising results of using multiple monitors places the pressure on HCI researchers to find answers for all the challenging HCI questions associated with their use.

It is well-known that the usability of the interface is a crucial factor in determining the user productivity. Therefore, interface designers try to design more usable interfaces on multiple displays with the intention to enhance user productivity. However, we believe that there are not enough interface guidelines available in the literature that target different types of secondary displays. This motivates us to start investigating and looking for answers to some of these questions.

1.3 Thesis Statement

The following statement captures and motivates the nature of this work:

> Individual and group performance differs when information is displayed on secondary personal displays as well as large shared displays, even at identical visual angles. Understanding these differences leads to the development of a set of guidelines that focuses designers' attention on opportunities to move information from secondary personal displays to large shared displays for increased performance of individuals and groups in collaborative environments.

1.4 Research Approach

In this work we combined different types of research approaches. Specifically, we combined theoretical work such as critical parameters with empirical evidence from user studies to understand the effects of using large shared displays as compared to secondary personal displays. Then, we designed controlled experiments to verify and understand additional effects of using large shared displays with a primary personal display. Based on these results, we used subjective evidence such as informal observations and interviews to develop design guidelines for large shared displays. Finally, we developed a system that follows the proposed design guidelines and conducted an in-situ user study to assess the validity of these guidelines, leading to more interesting observations about the system.

1.5 Contributions and Anticipated Impact

Following our research approach, we grounded our work on established theoretical work of Newman on establishing critical parameters for interface evaluations (Newman 1997). McCrickard et al developed a set of critical parameters for evaluating notification systems called the IRC Framework (McCrickard, Chewar et al. 2003). This framework is composed of three main critical parameters: interruption, reaction, and comprehension.

In this work, we look at secondary displays as notification systems. Hence, we assess different types of displays based on the IRC Framework. We chose two different types of displays that are used in most of the work environments; secondary personal and large shared displays. Based on the results, we started to develop interface guidelines for large shared displays and using them in a real life system implementation. The contributions of this work are presented below.

Comparison between personal and large shared displays in terms of interruption and comprehension

In this work, we present empirical studies demonstrating the differences between secondary personal displays and secondary large shared displays in terms of interruption and comprehension. The studies document a higher level of interruption when using large shared displays over personal displays. Similarly, a higher comprehension is also noticed, which shows promising opportunities for potential benefits through shifting peripheral information to large shared displays.

Interface design guidelines for large shared displays

Our investigation to the effect of large shared displays on users' performance in primary and secondary tasks shows that there are significant improvements on the secondary task performance. Users show remarkable preference to the use of large shared displays as opposed to secondary personal ones. Therefore, we delve more deeply into the science of design when large shared displays are used. We run empirical studies as well as qualitative research to develop and find interface design guidelines when using large shared displays in collaborative environments.

SuperTrack: A system for improving the group awareness in collaborative software development environments

To gauge the benefits of our research results in real life, we introduced large shared displays into a collaborative software development environment. A system aiming to improve group awareness in collaborative software development environments is implemented and presented on a large shared display. Qualitative investigations were conducted to study the effects and implications of employing our interface guidelines in real life. We see that our research results can have an important impact on designing real-world computer supported collaborative work environments. Specifically, SuperTrack demonstrates a system for enhancing group awareness in collaborative software development environments using large shared displays.

1.6 Overview of the Thesis

In this chapter, we briefly presented a high level description of the problem at hand, the motivation, our research approach, and contributions. In the remainder of this section, we describe the thesis organization.

Chapter 2 discusses related work that has contributed to a better overall understanding of large shared displays and how they can be used in our computing environments. In addition, the chapter explores the existing work that shows the importance of multiple monitor systems. Moreover, it presents the research previously contributed to designing user interfaces for secondary tasks on large shared displays. Finally, a discussion of the existing interface design guidelines for large shared displays is presented – showing the need to think about more design guidelines when large shared displays are used as notification systems.

Chapter 3 presents a series of experiments showing that large shared displays, even when viewed at identical visual angles with personal displays, affect the way we perceive certain information. The experiments do not only present performance differences and enhancements when large shared displays are used, but also compare the large shared displays to personal ones based on the critical parameters of interruption and comprehension.

Chapter 4 presents analytical and empirical studies leading to the development of a set of interface design guidelines for large shared displays. Specifically, a user study is explained and followed by a user interview about the characteristics of large shared displays and visual information design principles.

In chapter 5, SuperTrack, a software project tracking tool, is presented along with its motivation, implementation details, and experimental results. Moreover, the chapter applies the previously developed interface design guidelines to the real world.

Finally, in chapter 6, the work and contributions resulting from this research are summarized. In addition, we outline directions for future work.

Key materials from the main experiments in this thesis are included within the Appendices.

2

Related Work

Interestingly, we believe that there is a lack of design guidelines for large shared displays in collaborative environments. This chapter starts off with a discussion about the benefits of using multiple monitors instead of a single monitor. Then it presents the related work documenting productivity benefits for using large shared displays in various work environments. Showing the need for additional design guidelines, a description of the existing guidelines for large displays is also provided. Finally, we discuss the angle from which we look at large displays, notification systems, and present the previous research results in this regard.

2.1 Single Monitor vs. Multiple Monitors

The increased pace of technology advancements has substantially changed the way people live their life. Nowadays, people use information technology in a wide variety of aspects. For example, ordinary users check their e-mails, pay their bills, and search for information using their computers. Other users have different uses depending on their domain. However, it is not uncommon to even find users that run multiple tasks simultaneously. For example, users can be editing a document while attending a video conference lectures and monitoring stock prices at the same time.

With the increased computing capabilities, users become more ambitious to accomplish more tasks in less time. At the other extreme, HCI researchers try to investigate all possible ways to increase the productivity of users. Given the standard computing model as illustrated in Figure 1 some of HCI researchers investigate issues related to outputting and displaying information to users. Hence, there are different types of methods for displaying information such as printers, monitors, peripheral displays, and sound alerting systems. However, the traditional and most commonly used method of displaying information is the traditional computer monitor.

Figure 1: Standard Computing Model

With the increased number of applications and tasks that the user can run and manage simultaneously, users started to look into enhancements to their traditional desktop monitor. Henderson and Card introduced a window management solution called Rooms (Henderson and Card 1986). The Rooms solution depends on the idea of expanding the desktop space using multiple virtual desktops. Another solution that is being noticed lately is that users tend to use multiple monitors associated with the same computer. Although this might sound as a simple move towards increasing the display space, it started to interest a number of HCI researchers. Based on that, there are a lot of research results in the literature record productivity benefits for using multiple monitors instead of a single monitor.

Grudin studied the use of multiple monitor systems (Grudin 2001). Although that there were not quantitative research results, Grudin formed observations that show a substantial subjective benefits when using multiple monitors over single ones. Grudin also shows that users tend to move secondary tasks to the secondary monitors and hence to the peripheral attention. After three years, Ringel formed taxonomy of how users tend to organize their tasks on several virtual desktops (Ringel 2003). His research also shows that users prefer to use the extra space resulted from virtual desktops or physical multiple monitors.

Colvin et al conducted productivity testing on the use of multiple monitors as compared to personal monitors (Anderson, Colvin et al. 2003; Colvin, Tobler et al. 2004). The study involved 108 participants editing PowerPoint slide shows, spreadsheets, and text documents. The quantitative performance measurements as well as the usability testing results show that users are significantly more productive than using a single monitor. Czerwinski et al studied user productivity when using multiple monitors in complex and multiple window tasks (Czerwinski, Smith et al. 2003). Results indicate significant performance improvements when using multiple monitors.

Given that there are many benefits for using multiple monitors rather than a single monitor, there are many challenges that we need to face and questions that need answers. For example, Hutchings and Stasko in their work discussed the problems associated with the existing window managers because they were not designed for multiple monitor systems (Hutchings and Stasko 2007). Specifically, they focused on the problem of unpredictable behavior of dialog boxes in multiple monitor systems. Ringel in his research shed the light on the fact that existing applications are not designed to be used in multiple monitor systems (Ringel 2003). Among the design implications that Ringel focused on is that maybe the applications need to remember which monitor it was displayed in. Booth et al focused on the collaboration problems among multiple monitors (Booth, Fisher et al. 2002). Dudfield et al studied the human factors issues of large shared displays in military applications (Dudfield, Macklin et al. 2001).

This shows the emergent need to further study the existing applications and identify the shortcomings when being used in multiple-monitors environment. Furthermore, we believe that there is an urgent need to find design guidelines and frameworks that interface designers can use to design applications that utilize multiple monitors. The question becomes more challenging when we see that users tend to use different types of displays in their multiple monitor system. For example, a personal monitor, a large monitor in their work environment, and a small PDA display for his quick personal information. In the following section we discuss the different impacts when using different types of displays in a multiple monitor system. Specifically, we discuss the effects of using large shared displays as compared to having secondary personal monitors.

2.2 Large displays

With the increased use of multiple monitor systems, users started to use more than 2 monitors which shed the light to forming large displays. Tan gives a good overview of engineering large displays as well as descriptions of many examples of the use of large displays (Tan 2004). Although large displays by definition consume a large space from the workspace, researchers documented tremendous benefits for using them.

Large displays have been used in several work environments. For example, Elrod et al designed Liveboard, a large interactive displays system to be used in group meetings, and presentations, and remote collaboration (Elrod, Bruce et al. 1992). Ganoe et al developed Classroom BRIDGE system which uses large displays in educational environments (Ganoe, Somervell et al. 2003). Biehel et al developed a visualization that is intended to enhance productivity when using large displays in collaborative software development environments (Biehl, Czerwinski et al. 2007).

Czerwinski et al studied the productivity benefits when using large displays as compared to small personal monitors (Czerwinski, Smith et al. 2003). Quantitative research results indicate that users are significantly more productive when using multiple monitors forming large display than using a single monitor. Moreover, users are more satisfied accomplishing complex tasks using large displays than personal monitors. Ni et al studied the use of large displays in information-rich virtual environments (IRVE) (Ni, Bowman et al. 2006). Results show that task performance in IRVE is improved with increasing the display size as well as the resolution.

On the same vision, Tan et al studied spatial performances between desktop and large displays with the same visual angle (Tan, Gergle et al. 2003). Their study focused on the individual spatial performance rather than group performance. Results show that users perform better when large displays are used for spatial orientation tasks. Moreover, their results suggest that large displays provide users with greater sense of presence.

Given the qualitative and quantitative results showing the productivity benefits of using large displays, there are different types of problem associated with using large displays. User interaction with large displays differs than interaction with personal ones (Swaminathan and Sato 1997). Therefore, several researchers started to study different novel interaction techniques that suit the applications for large displays. For example, Jiang et al studied the use of the cameras equipped in handheld devices to intuitively interact with large shared displays (Jiang, Ofek et al. 2006).

Most of large displays are used to share information in work environments. Therefore, there are issues regarding personal information privacy associated with using large displays in shared environments. Tan and Czerwinski prove that even with the same visual angle users can face personal information privacy problems with using physically large shared displays (Tan and Czerwinski 2003).

Large displays suffer from several usability problems. For example, window management problems resulting from the notifications and dialog boxes that pop up in the large display (Hutchings and Stasko 2007). Moreover, users tend to accomplish more tasks when using large displays hence, using more windows which triggers task management problems. These problems associated with using large displays attracted the attention of HCI researchers to study the different types of impacts of using large displays.

Although there are several research results documenting productivity benefits of using large displays as compared to personal ones, there is a very limited research comparing large displays to secondary personal displays. Therefore, we see a need to understand and compare the different benefits as well as problems between using large displays and secondary personal ones.

2.3 Existing design guidelines

Once HCI researchers realized the various benefits associated with using multiple monitors and large displays, they realized the need for design guidelines and frameworks for designers. On the other hand, designers expect to be able to use the provided design guidelines and frameworks to produce highly usable and productive interfaces for multiple monitors and large display environments. Therefore, there are many research efforts directed towards analyzing large displays and developing design guidelines.

Starting with design languages, researchers believe that the existing design languages do not support the design requirements for large shared displays or multiple monitor systems. In an attempt to develop a design framework for designers, Goldberg and Mochel propose a new design language that takes advantage of the extra space provided by large displays (Goldberg and Mochel 2006). The new proposed language is supposed to focus on sharing design components and designing simultaneous interfaces on multiple monitors.

Su and Bailey investigated the impacts of different configurations of the physical positioning of multiple large displays on users (Su and Bailey 2005). Twenty users participated in their study of manipulating the configurations of two large displays: physical separation, angle between them, and symmetry when facing each other. Research results show that different configurations impact the user performance in the given tasks. Three guidelines were formed regarding positioning multiple large displays: (1) the displays should be placed in a horizontal visual angle that does not exceed 45°; (2) users should be facing the large displays; (3) The angle between large displays should not exceed 45°.

In their Courtyard system, Tani et al used large shared display among team members with a personal one for each team member (Tani, Horita et al. 1994). They developed a set of interface requirements needed for designing complex real-life systems. These requirements include: (1) the interface should support easy transitioning between the general view on the large display and the detailed view on the personal displays; (2) the interface should support quick retrieval of detailed information on the personal displays.

2.4 Large displays as notification systems

Evaluating user interface designs has always been a challenging question for HCI researchers. Newman assessed the benefits of using critical parameters in designing interactive systems (Newman 1997). Critical parameters are defined as quantitative measures of performance that can be used to evaluate the ability of a design to serve its purpose. With critical parameters, designers can compare different designs and determine the most appropriate design for a specific purpose. To enhance design productivity, Newman shed the light on the need to find a set of critical parameters for designing interactive systems (Newman 1997). Based on that, many researchers started to look into the possibility of developing a generic set of critical parameters for generic domains of applications.

Notification systems is one of the domains that can be enhanced using critical parameters evaluations. Chewar and McCrickard define notification systems as "software and physical interfaces that deliver information of interest to users engaged in other activities" (Chewar and McCrickard 2003). In an attempt to find critical parameters for evaluating notification systems, Chewar et al identified 3 different critical parameters called IRC framework (Chewar, McCrickard et al. 2004): interruption, reaction, and comprehension. To assess the benefits of this framework, Chewar et al designed LINK-UP, a system for automating the evaluation of notification systems critical parameters (Chewar, Bachetti et al. 2004).

Interruption. In the context of notification systems, Chewar identifies it as reallocation of users primary attention from the primary task to a notification (Chewar 2005). Depending on the nature of the situation, high or low levels of interruption may be desired. For example, high levels of interruption are desirable in highly critical command and control systems. However, low levels of interruption are more desirable in in-vehicle information systems. Therefore, identifying the appropriately needed level of interruption would help design more useful notification systems.

Reaction. Chewar relates to reaction as the response to the stimuli provided by notification systems (Chewar 2005). Depending on the type of notification system, users may be required to react to the notification as quick as possible. For some systems such as military systems, users are required to respond to the notification as fast as possible to save lives. However, users are not required to be highly responsive to email notification for instance. Therefore, it is believed that understanding the required level of reaction has a crucial impact on the success of designing a notification system.

Comprehension. Chewar refers to comprehension as the amount of information that is retained into memory after responding to the notification system (Chewar 2005). Many researchers in human-factors and ergonomics studied the effects of visual characteristics such as text and color on memory. Chewar et al found that understanding the needed awareness of the information provided by the notification affects the level of comprehension (Chewar, McCrickard et al. 2004).

To use one of the critical parameters benefits, Chewar classified the types of notification systems based on the values of critical parameters (Chewar 2005). Each of the critical parameters takes a value between 0 as low and 1 as high. In her classification, she used the extreme values of each of the critical parameters to identify a domain of notification systems. Chewar's full classification of notification systems families is provided in Table 1 (Chewar 2005).

Table 1: Notification systems classification based on IRC Framework values

IRC Values	Notification Families
001	Ambient Media
010	Indicator
011	Secondary Display
000	Noise
100	Diversion
110	Alarm
101	Information Exhibit
111	Critical Activity Monitor

In this thesis we study large displays as notification systems. Chewar classifies secondary display notification systems as low interruption, high reaction, and high comprehension. However, as previously mentioned, there are different types of secondary displays including large shared displays. Would Chewar's classification of secondary displays still be valid when secondary displays are actually large shared displays? In this work, we focus on studying the interruption and comprehension associated with large shared displays.

3

Secondary Tasks: How Close Should They Be?

This chapter presents an experimental analysis of the costs and benefits of using personal peripheral displays in comparison with shared or public ones, highlights the interruption and comprehension tradeoffs associated with the use of either of the peripheral displays, and emphasizes the need for design guidelines for large peripheral displays to maximize user performance in collaborative environments. Specifically, two empirical studies are presented. The first experiment tends to study the different effects of using personal and public peripheral displays on the primary and the secondary tasks. The second experiment is used as a proof of concept, since we changed the primary and secondary tasks to illuminate the effect of the tasks themselves. Results of the two experiments show that there is a significant performance improvement on the secondary task associated with the use of large shared displays as opposed to personal ones. The primary task performance degraded in both cases and we provide an explanation for this degradation. Finally, the preferences of the users tend to show promising opportunities to shifting information on large shared displays.

3.1 Introduction

With the increased diversity of computing display capabilities, users become more ambitious in undertaking multiple tasks simultaneously such as editing a document while attending a videoconference, monitoring email, and instant messaging. Multiple tasks are also seen in critical scenarios such as command and control situations and traffic monitoring. Generally, in dual task situations, the user works on a primary task while attending to one or more secondary task displays. With the addition of each secondary display comes the possibility of interruption of the user and degradation to primary task performance. For example, classroom environments have integrated multi-monitor displays to allow students, working projects, to maintain awareness of each other’s tasks and schedules while working in a primary task on a desktop computer (Ganoe, Somervell et al. 2003). The introduction of multi-monitor systems in classroom environments needs to be further explored to maximize the utilization of the user’s attention to both the primary and secondary tasks.

Interruption is defined as an event prompting transition and reallocation of attention focus from a primary task at hand to the notification. Researchers explored various possible effects of interruption on the user’s performance, memory, and attention. Our research theme investigates how interface design decisions for secondary displays affect the allocation of user attention to optimize user performance on both primary and secondary tasks. Specifically, this chapter explores the costs and benefits of public secondary displays, such as projected videoconference or lecture displays, and private secondary displays, such as secondary monitor or second personal computer, to understand the resulted interruption and comprehension of the different displays and their impact on the user’s primary and secondary tasks.

Our approach in learning about the costs and benefits of public and personal secondary displays is to conduct various empirical studies. First, we want to consider if there is any impact on the primary and the secondary task performances with the use of peripheral display. Moreover, we want to explore and compare the personal as well as the public peripheral displays in terms of interruption and comprehension. Therefore, in the following we present with two experiments to help us do this comparison and study. Finally, a discussion of experimental results is presented.

3.2 Experiment 1

Our first study is motivated by the need to understand the effects of using personal and public peripheral displays in various work environments on the performance of primary and secondary displays. As we discussed IRC framework to evaluate the design objectives of notification systems, we need to evaluate the use of large shared display as a notification system to be used for individuals or groups. We tried to illuminate the "R" component since it would require direct interaction between the user and the display which would have added another dimension to the complexity of the study. Therefore, in this study we explore the "I" and "C" components.

3.2.1 Hypotheses

In this experiment we hypothesize that, in dual-task situations, using large shared display for displaying the secondary task leads to balanced attention among tasks and hence performance improvements.

a) Using large peripheral display would enhance primary task performance rather than using a personal peripheral one

b) Using large peripheral display would enhance secondary task performance rather than using a personal peripheral one

3.2.2 Experimental Design

To gauge the differences between personal and public displays, we compare them based on two aspects: *interruption* and *comprehension*. Interruption is measured based on the degradation in the user's performance when working on the primary task and comprehension is measured based on the amount of information from the secondary task that is retained by the user. Specifically, to test our hypothesis we designed and conducted an empirical study. This study demonstrates a dual-task scenario where users are required to achieve high performance in both primary and secondary tasks. The experiment explores three conditions: a base condition, where each participant is engaged in primary task without any interruption from secondary tasks to measure primary task performance degradation; a personal display condition, where each participant is not only engaged in a primary task, but also in a secondary task displayed on a personal secondary display; and a public display condition, where the secondary task is performed using a projected public display. The display size for both personal and public displays is kept the same through maintaining a constant visual angle that facilitates the same field of view for both displays. The experiment used a within-subject design for the rounds, with display type changed via a Latin Square design.

The environment is set in such a way that two monitors are close to each other with a slight angle between them that makes both of them easily seen by the user. To test the effect of only one factor, we fixed the field of view. Therefore, the secondary large projected display is in the same field of view as the secondary personal one as shown in Figure 2.

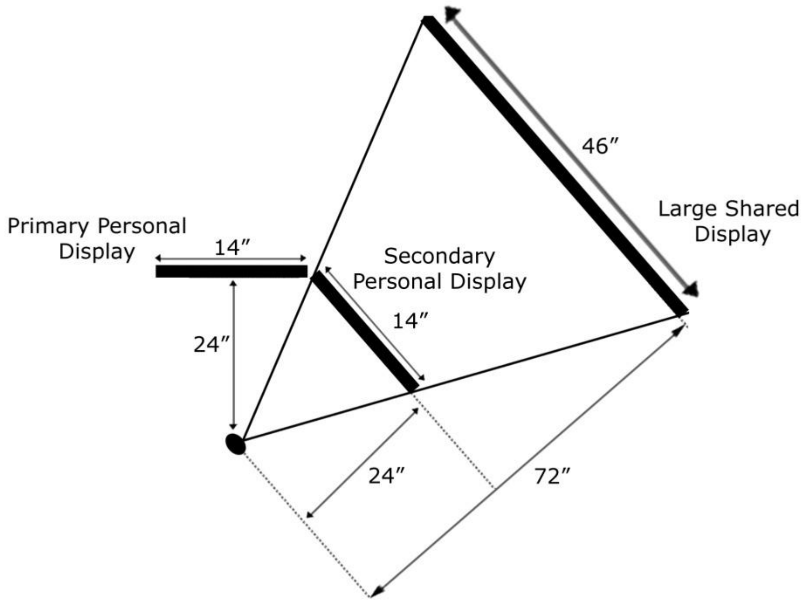

Figure 2: The setup of the primary and secondary displays. The personal display (close) and the large display (far) are in the same field of view

The experiment involves a questionnaire for each participant at the end. The questionnaire is designed in a manner such that it indicates the user's preference for conditions, perceived interruption, difficulty of primary task, and the difficulty of questions about the pictures at the end of each round.

3.2.3Participants

Thirty subjects voluntarily participated in our experiment. Participants were all either computer science students or graduates, which indicates that they are good computer users. We screened users to be fluent in English and to have normal or corrected-to-normal eyesight. The average age of participants was 24.5, ranging from 19 to 30 years of age. Students were provided with extra grades for participation.

3.2.4Materials

We used two PCs have the following specifications. Each of the PC has Intel Pentium IV 1.8 GHz, 256 MB RAM, and 17” ViewSonic flat monitor. All displays were configured to work in a resolution of 1024 x 768 and were calibrated to be roughly equivalent in brightness and contrast. The large public display is a projection obtained through a Philips projector on a wall. However, the projected display is in the same field of view as the secondary personal display as shown in Figure 3.

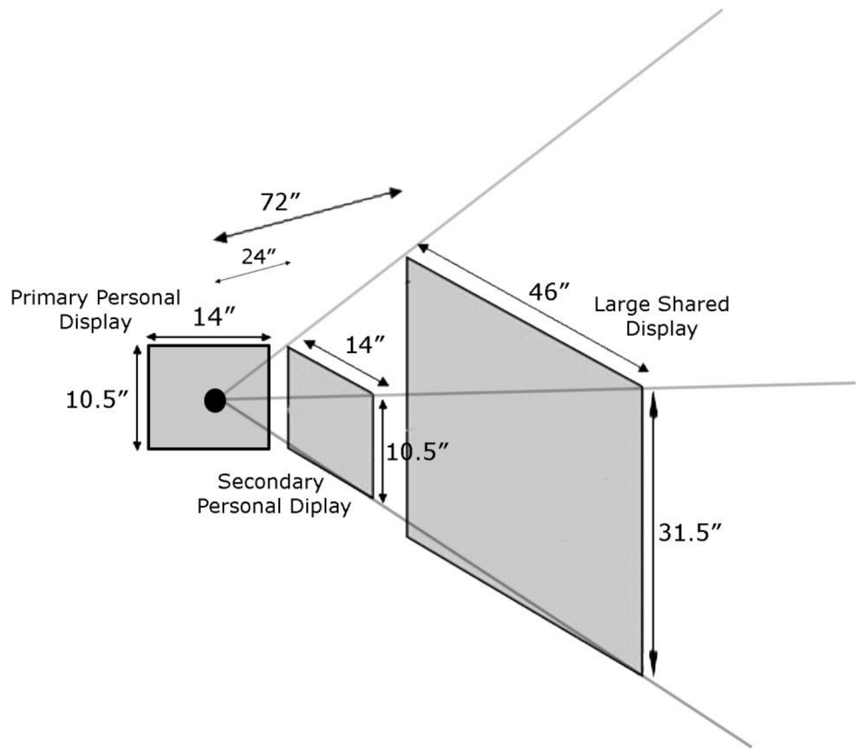

Figure 3: The large shared display is a projected display in the same field of view as the personal display

3.2.5Tasks

The primary task is a typing task in which participants are given a set of articles that are required to be retyped in a text file. All of the articles are piled up together in one PDF file. This file is horizontally top aligned in the primary display. Underneath the PDF file an empty txt file is opened in Microsoft Notepad. Each participant is given 3 minutes and is asked to type as many correct words as possible. A threshold of mistakes is defined to be a ratio of 10% of misspelled words to correct words. If a user exceeds this threshold, he or she gets expelled from the experiment.

The secondary task asks participants to look at pictures on either a personal or public display. Each picture is shown for duration of 15 seconds with no lag time between each picture. Participants are asked 12 questions at the end of each round about the quickly perceived visual properties in the pictures displayed in the round. Questions are single choice questions where only one answer is correct.

The experiment involves a number of variables. The independent variables are the two types of secondary displays (personal, public) and the articles to be typed. The dependent variables are the interruption measured by the degradation of the number of words typed throughout the time limit of 3 minutes and the comprehension measured by the number of correct answers to the questions at the end of each round about the pictures displayed in the round.

3.2.6Procedure

After a background survey is filled out, a set of instructions is given to participants in order to explain the tasks required from them throughout the experiment. In the base condition, users were encouraged to maintain a good performance in the primary task. In the other two conditions, users were encouraged to maintain good performance in both the primary and the secondary tasks.

Participants start with the base condition in which they try to type as many words as possible in the time limit of 3 minutes. Participants then, using Latin Square design, follow with the other two conditions in which they are required to type as many words as possible during the time limit of 3 minutes, while maintaining awareness of the secondary task on the secondary display. At the end of each round, participants are asked a few questions about the pictures displayed in the round.

By the end of the experiment, participants answer a questionnaire about their preference among the conditions presented and the difficulty of each of the tasks. Participants were encouraged to provide opinions on their answers and any feedback about the experiment. The experiment takes around 17-20 minutes with each participant.

3.2.7Results

The results from this experiment are represented in three parts. First, the performance on the typing task is explored, then the performance on the picture comprehension (secondary) task. Finally, an investigation of the preference measures collected at the end of the study is presented.

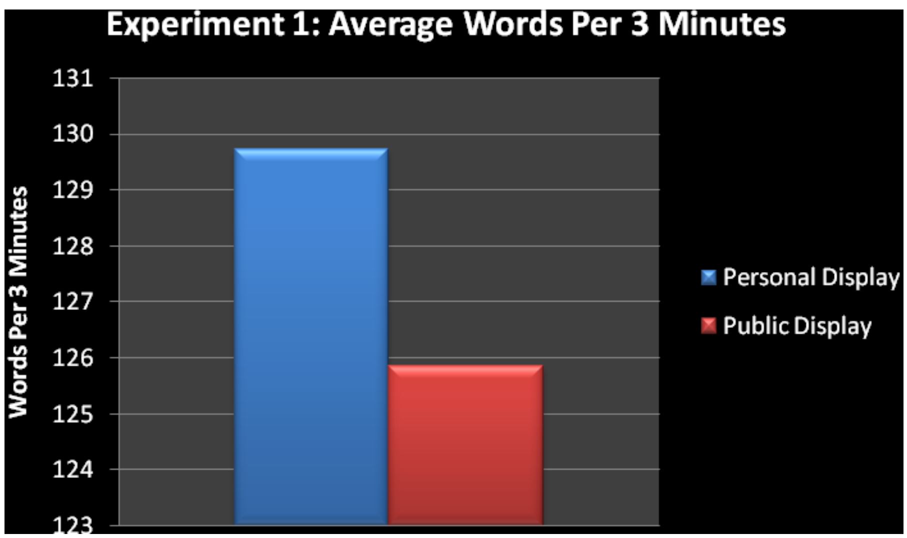

Figure 4: Main effect of secondary display type. Users performed significantly better when personal display was used rather than large shared display

Primary Task Performance

The data of the primary task was analyzed at the summary level. The dependent variable, average of words typed per 3 minutes, is analyzed with repeated measures analysis of variance (RM-ANOVA). A significant main effect of the peripheral display was found on the performance of the primary task (p=0.0006) with the personal peripheral display resulting in a higher words per 3 minutes than the public on average (129.7 vs. 125.8, respectively) as shown in Figure 4. Although the performance of the primary task is better when personal peripheral display is used, this performance is less than when there wasn't a secondary display (the base condition).

Overall, a significant improvement in the performance of the primary task was found on users working on personal peripheral displays. Keep in mind that while the absolute size of the personal peripheral display is smaller, the perceived or retinal display size is kept nearly constant with the same field of view. One way to explain the performance improvements on personal peripheral displays is the higher interruption resulting from the use of large peripheral display. Throughout the experiment, we noticed that some participants didn't even notice that the picture changed on their personal secondary display, while we didn't notice such an oblivious behavior on the large peripheral display.

Secondary Task Performance

In the secondary task, the data was also analyzed at a summary level. The number of correct answers was the dependent variable. T-Test was used to test significance of results. A main significant effect of the peripheral display type was found on the performance of the secondary task (p=0.003) with the public display resulting in more correct answers than the personal on average (11.1 vs. 10.2; respectively).

Although the performance of the primary task was better when personal peripheral display was used, this isn't the case on the secondary task. Conceptually, the dependent variable for experiment 1 represents the comprehension level of using a larger peripheral display. Therefore, we believe that there is higher comprehension component resulting from the use of large peripheral displays than personal ones.

Preference Data

In addition to the performance data, preference data was gathered at the end of the study. The questionnaire assesses user preference to the different secondary displays, and explores what users did and did not like about each secondary display. Based on our study of users' answers to the questionnaire, we found users tend to prefer the use of large secondary displays rather than personal ones. In accordance, most of the users mentioned that the use of a large display in the secondary task made it easier for them. However, a small number of users, four, stated that they prefer the use of personal display as it makes them feel not exposed.

3.2.8Summary

This experiment was designed to study the interruption and comprehension resulting from the use of different peripheral displays. The results show significant performance improvements of the primary task when a personal display is used and show significant performance improvements of the secondary task when a large shared display is used as opposed to personal one. Users tend to prefer to use the large displays since they believe it is much easier for them to get notified of changes and such when a large display is at hand. The ease of seeing was also one of the major points mentioned by users.

Going back to the overall picture of our thesis, we can conclude that there is a higher interruption and comprehension components from the use of large shared displays. These higher components may give an opportunity to serve as a guideline for designers to consider when designing interactive workspaces that use large shared displays.

Looking back at the nature of the secondary task chosen, we found that the secondary task was simple in term of visual perception properties. The questions at the end of each round tackled quickly perceived visual properties which were ordered by McCrickard et al's work (Tessendorf, Chewar et al. 2002). The question popped up is that would these results hold in a more realistic dual-task scenario. Therefore, we designed a second similar experiment that would have the same objectives with the exception that the tasks are more realistic ones. The second experiment should guide us to further investigate the performance improvements resulted from the use of secondary public displays as compared to personal ones.

3.3Experiment 2

This experiment is designed to further investigate the interruption and comprehension components when personal and large shared displays are used in the work environment. Since we are interested to know if our results would hold in the real-life situations or not, more realistic dual-task scenarios are the main focus of this experiment. We designed the experiment to investigate in software development situations.

Based on our study of various applications of HCI research, we found that there is a considerable research effort on designing source code visualizations that would help software developers to be more productive in their development tasks. Therefore, we tackle the dual-task scenario of programming while using source code visualization. The primary task of this scenario is the programming task while the secondary task is using the source code visualization to help the programmer to boost his performance.

3.3.1Hypotheses

As stated before, this experiment is similar to the first experiment. Therefore, it has the same hypotheses. To be more specific, we hypothesize that there will be performance improvements on the secondary task due to the use of secondary public displays even with a secondary task that was primarily designed to be a primary task such as source code visualizations for code analysis. As for the performance of the primary task, we hypothesize that there will be performance improvements when a large shared display is used as opposed to secondary one.

3.3.2Experimental Design

To further investigate the reasons for performance improvements when employing secondary public displays, we conducted an empirical user study. This experiment is similar to the previous one in terms of the nature of the conditions. The study explores three conditions; a base condition in which the user works on a primary task without any interruption from secondary tasks; a personal display condition in which the user works on a primary task yet engages in a secondary task displayed on a secondary personal display; a public display condition where the secondary task is displayed on a secondary public display. The environment setup is similar to the previous experiment.

This experiment also involves a questionnaire at the end of the experiment for each participant. This questionnaire is designed in a way to test the user preference to each condition. The questionnaire is also similar to the one in the previous experiment. However, we modified the questions to fit the nature of the experiment and the new designed tasks. Other questions were added to further illustrate the difficulty of the primary as well as the secondary task and the attention required to successfully engage in the secondary task.

3.3.3Participants

Thirty participants volunteered for our study. The participants were ten teaching assistants with excellent programming skills and twenty students who were intermediate to experienced programmers using C++. We screened users to be fluent in English so that they could easily understand the tasks required as well as the instructions for the experiment. The average age of all participants was 20.7 (23.5 for teaching assistants, 17.9 for students), and the participants ranged from 16 to 28 years of age. Students were offered extra grades for participation and teaching assistants were paid.

3.3.4Tasks

The primary task is designed to be a programming task in which each participant is required to do some code modifications. Three existing free C++ source codes of almost the same complexity were employed in the primary task where each of them is composed of multiple files of source code. The programming tasks for these source codes were either adding simple new functionality to the system or simply modifying an easy piece of code. Each of these tasks was estimated to be finished in time less than 12 minutes. Therefore, a time limit of 12 minutes is applied for participants.

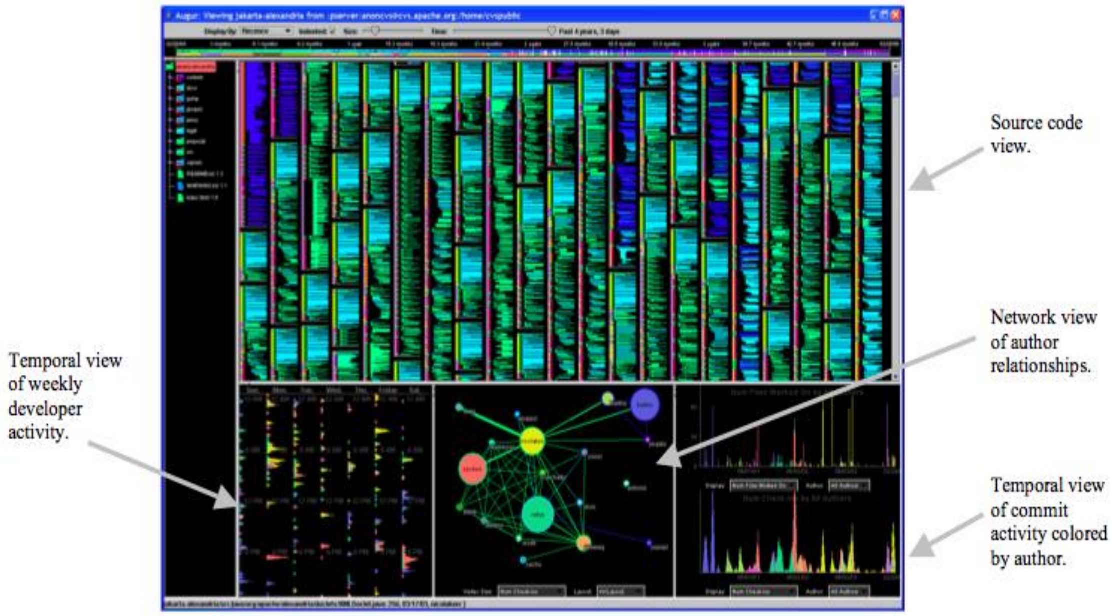

Figure 5: Augur, line based source code visualization, showing various views including source code view, network view of author relationships, temporal view of commit activity colored by author, and temporal view of weekly developed activity. Used with permission of author, Jon Froehlich, University of Washington, 2004.

The secondary task is monitoring and interacting with source code visualization. A question will be manually given every 3 minutes to a total of 2 questions about code analysis. These questions can be easily answered using the source code visualization. Users can also use the source code visualization to further understand the source code at hand which should help them finish their programming task faster. The recruited source code visualization is Augur (Froehlich and Dourish 2004) which provides a line-based view of the source code such as Figure 5.

This experiment involves a number of variables. The independent variables are the two types of displays (personal, public), the source code visualization, and the three pieces of source code. The dependent variable, the variable to be measured, is the user performance in terms of timing, completeness, and correctness.

3.3.5Procedure

All participants are asked to fill out a background survey. Two instruction sets were given for each participant; one explains the tasks required in the experiment; the second one explains how to interact with Augur. In base condition, participants are given a folder of source code and are given a written task on a sheet of paper which participants should accomplish. These tasks are simple source code modification tasks or simple functionality implementation. Each participant is timed until the task is accomplished, so that we can monitor and spot the performance improvements that occurred in one condition over the other. The participant is asked two code analysis questions, one every 3 minutes. During the other two conditions, the participant can use the source code visualization, Augur, on the secondary display to help him understand the code at hand quickly, and to answer the code analysis questions. Augur was displayed on a secondary display and the interaction with Augur was through an additional mouse other than the one used in the primary task. The source codes were installed on a source control server so that Augur automatically updates itself with the modifications of source codes.

At the end of the experiment, all participants are given a questionnaire about their preference as to the conditions involved. Moreover, participants are encouraged to provide their opinion and feedback about the answers and the experiment. On average, the experiment took around 45 minutes from each participant.

3.3.6 Results

Primary Task Performance

The dependent variable is the time (in seconds) elapsed to accomplish the programming task. The data was analyzed using RM-ANOVA. No significance was found in the primary task performance data in personal or public secondary displays. Our rationale is that the interruption caused by the secondary display caused no distraction to users when thinking about their programming task. Therefore, we believe that there is an interaction between the primary task and the peripheral display.

Secondary Task Performance

Similar to the first experiment, the data was analyzed at a summary level and T-Test was used to check for significance of results. The dependent variable for the secondary task is the time (in seconds) elapsed to answer the code analysis question. A significant difference is found when users answer the questions ($p < 0.0001$) with lower average time on public peripheral display as compared to personal one. No significance was spotted in the correctness of the answers as almost all participants were able to answer the questions correctly and in the time span of 3 minutes.

Preference Data

Similar to the results of preference data we gathered in Experiment 1, we found that users tend to prefer using large displays rather personal ones for their secondary tasks. Users noted in their comments that large displays made it much easier for them to analyze the code, since it gave them a better and easier look to their source code.

3.3.7Summary

In this experiment, we found no significant performance difference on the primary task caused by the use of peripheral displays. However, we found a significant secondary task performance improvement when large peripheral displays were used as opposed to the personal ones. Similar to the first experiment, we found that participants prefer to use large peripheral displays to manage their secondary tasks instead of personal ones.

3.4Conclusion

In this chapter, we introduced two empirical studies to compare personal and public secondary displays. The comparison points include the interruption and comprehension components, the primary task performance differences, and the secondary task performance differences. The first experiment used lab-based primary and secondary tasks. The results of the first experiment show that there is a significant performance difference of the primary task in a way that when large shared display is used, the performance of the primary task is better than when personal secondary display is used. Also, it shows that there is a significant performance improvement in the secondary task performance. The second experiment was designed to target more realistic dual-task situations. Therefore, the second experiment demonstrated the use of secondary displays in a programming environment. The results of the second experiment show that there is no significance performance difference with either of the displays. However, there is a significant performance improvement of the secondary task associated with the use of large shared displays.

The results of both experiments emphasize the importance of using large shared displays in various work environments. Therefore, we see the need for interface design guidelines for visualizing information in secondary public displays. Moreover, we need to study the tradeoffs associated with each of the guidelines. Thus, in the following chapter we investigate and explore guidelines for sharing and designing information in secondary tasks to be displayed on a secondary public large display.

4

Information Design Guidelines

This chapter emphasizes the importance of using large peripheral displays in collaborative environments, introduces the concept of source code visualizations, explains two different source code visualizations (VCN, Augur), presents an empirical study to investigate the differences between VCN and Augur, and argues two different design guidelines for how and when to use large peripheral displays in collaborative environments. The first guideline emphasizes the high interruption of using large shared displays. The second guideline tackles the problem of promoting group knowledge among the team members. Finally, the chapter presents a conclusion of the experiments results and the guidelines.

4.1 Introduction

As we have presented in the related work section, the use of large shared displays in various work environments is getting prevalent. Therefore, in the previous chapter, we studied the differences between large shared displays and secondary personal ones in terms of the interruption results from each one, the comprehension that users can gain from each one, and the performance effects on the primary and secondary tasks. After running two experiments in the previous chapter, we concluded that there is a higher interruption and comprehension factors associated with the use of large shared displays. Moreover, the performance of secondary tasks gets improved when it is presented on a large shared display.

Based on our results, we see a crucial need for designers to understand how and when to use large shared displays in collaborative environments. This urgent need for understanding how to design interfaces on large shared displays in collaborative environments was the motivation deriving us to further investigate interfaces on large shared displays. We started exploring various classifications of collaborative environments with the intention to apply our research on one of them, which is software development collaborative environment, where a number of developers work collaboratively to produce a software system successfully.

In earlier research, we used source code visualizations as the secondary task to study the effects of using large shared displays in more realistic dual-task situations. These source code visualizations are designed to help in program understanding and analysis, which is a critical part in the software development life cycle which involves collaboration among several coders and designers working together with millions of lines of code. To serve the purpose of better program understanding and faster collaboration and development, all developers could share a secondary large public display with the source code visualization displayed on it, while performing their primary development tasks on their personal computers. This setup will be used to further investigate the guidelines for designing interfaces on large public displays.

In the rest of chapter, two source code visualizations are described followed by an experiment to further understand the differences between these two source code visualizations. This is followed by a qualitative study, an interview with interface design professionals, to understand, analysis, and develop interface design guidelines on large shared displays. Finally, a conclusion is drawn.

4.2Source Code Visualizations

Source code visualizations started off as line-oriented code representation colored by code attributes and metrics. Eick et al. first proposed this idea in the system Seesoft (Eick, Steffen et al. 1992), a line oriented visualization tool, and was enhanced by several tools. Among those tools are: Augur, which is “a visualization tool that allows distributed software development. It creates visualizations of software artifacts, software development activities, and relationships among authors” (Froehlich and Dourish 2004); Aspect Browser, which is a tool developed using the map metaphor to track software evolution (Griswold, Yuan et al. 2001); sv3D, which used a 3D representation for the lines of code in order to compact the screen space needed (Marcus, Feng et al. 2003); Tarantula, which was developed primarily to aid the debugging process, by offering visualizations to help the debugger locate faults (Jones, Harrold et al. 2002); Gammatella, which offers three levels of visualization, statement level, file level, and system level (Orso, Jones et al. 2004), and ALMOST (Renieris and Reiss 1999).

When it comes to source code visualizations, we believe that none of the existing source code visualizations would efficiently maximize the utilization of software developers in collaborative software development environments, where the visualization is on a large shared display. Therefore, we need to compare existing source code visualizations. Based on what we get from these comparisons, we would start using them in a collaborative environment to further explore guidelines related to the use of large information displays in collaborative environments.

We set some criteria for choosing the visualizations we use in our experiments. To begin with, we need two source code visualization systems in order to perform some sort of feature comparison and be able to make decisions among the different design principles. Moreover, we need the two systems to be different in terms of several characteristics. Among these characteristics are the following:

- Data representation
- Interaction methods

- Display modes
- Multiple views
- Attention management

The two systems chosen to perform the investigation, according to the above criteria, are the Visual Code Navigator (VCN) (Lommerse, Nossin et al. 2005) and Augur (Froehlich and Dourish 2004).

4.3 SYSTEM 1: VISUAL CODE NAVIGATOR (VCN)

The VCN system (Lommerse, Nossin et al. 2005) is a toolset consisting of a set of three interrelated visual tools, developed for exploring large source codes from three different perspectives. The VCN is designed with the primary goal of helping developers understand the structure and evolution of programs.

4.3.1 Implementation

Every project P used by VCN consists of a set of source files, and for every file F_j, an annotated syntax tree with all constructs of F_j, along with several versions V_{ij} for every file F_j, for every version, attributes such as creation date and author are also recorded (Lommerse, Nossin et al. 2005).

4.3.2Views and Interaction

Shaded cushions are used to represent the hierarchies present in the data model for several reasons. First, shaded cushions are efficient in representing hierarchies on a single screen (Wijk and Wetering 1999; Voinea, Telea et al. 2004; Voinea, Telea et al. 2005). Second, cushions combine best with two-dimensional spatial layouts; cushions can be combined with treemaps to make the most out of the available screen space. Third, cushion rendering methods allow for interactive zoom and pan in the views. Finally, color encoding is used to display different code attributes via cushions.

(1)Syntactic View

To show the syntactic constructs in a given file, a cushion is defined whose geometric outline encompasses the construct's text. The cushions are color encoded to show the type of syntax construct they encompass. The code itself can be displayed as text inside the cushions, to allow for some level of detail that the user can control by changing the font height. The syntactic view is shown in Figure 6.

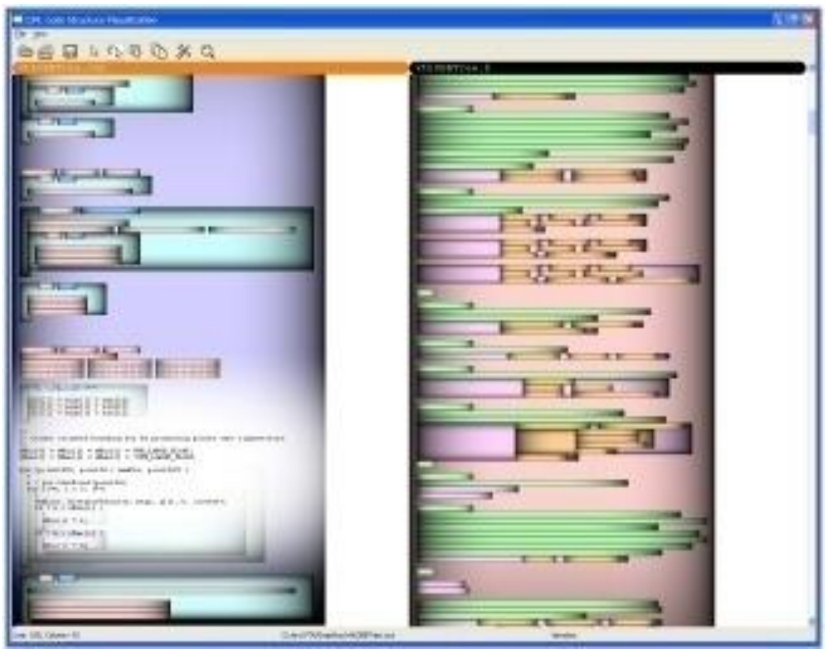

Figure 6: VCN Syntactic View. Used with permission of author, Alex Telea, Eindhoven University of Technology, 2005.

Another issue is navigation in the syntactic view. Code overviews can be generated by zooming out, by decreasing the font size, by fading out, and by decreasing the text opacity making some parts invisible. Details are provided on demand by displaying the code under the mouse in a text editor view. To overcome disruption of the navigation process, two cursors are provided, the spotlight cursor and the syntax cursor.

(2)Symbol View

The symbol view is a treemap showing all the symbols in a project along with their scopes. Each node in the treemap is colored to indicate its type. Interaction with the treemap is through brushing. For example, brushing over a file shows all symbols defined in that file as highlighted cushions. The symbol view is shown in Figure 7.

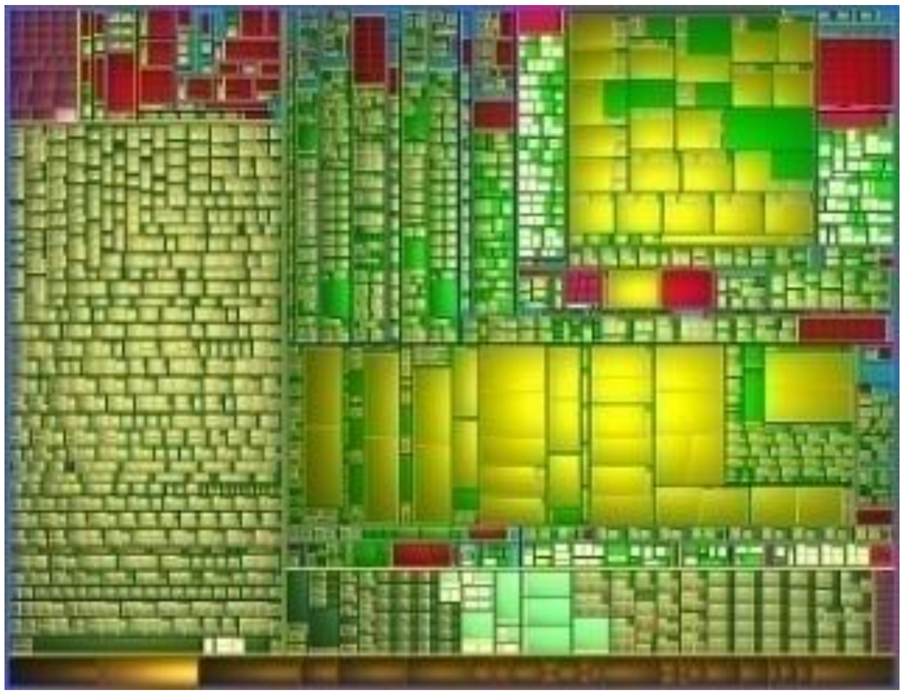

Figure 7: VCN Symbol View. Used with permission of author, Alex Telea, Eindhoven University of Technology, 2005.

(3)Evolution View

This view, shown in Figure 8, uses a pixel-filling display based on the file layout. For a file, the *x* axis maps the version number and the *y* axis maps the line number. A matrix is used for this visualization, in which every row displays a file and every column displays an attribute, such as line type and author. Color coding is used for different attribute types.

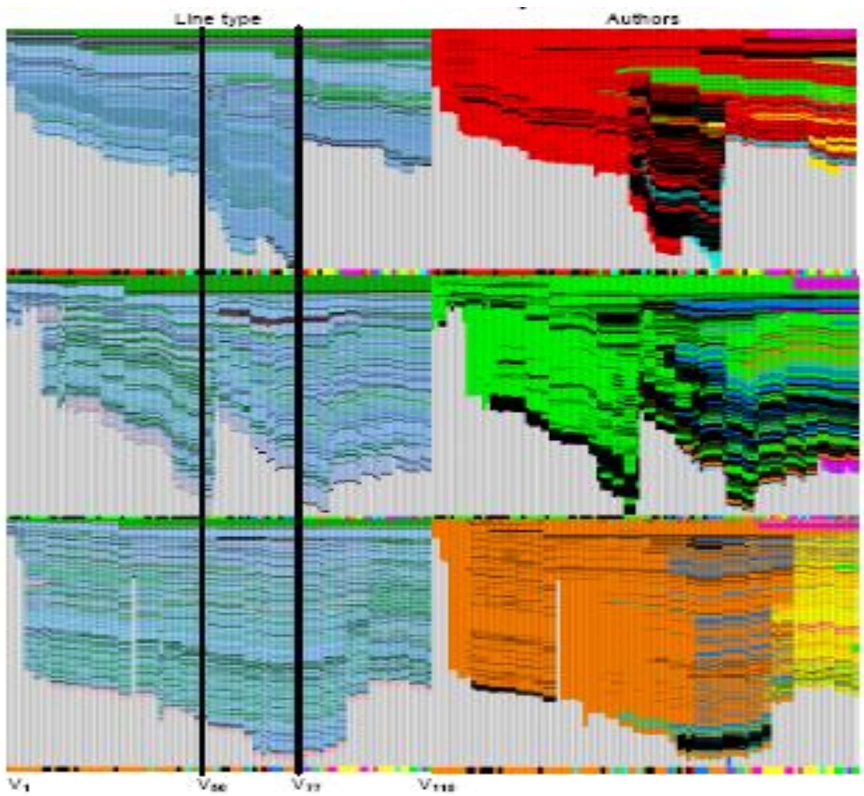

Figure 8: VCN Evolution View. Used with permission of author, Alex Telea, Eindhoven University of Technology, 2005.

4.4SYSTEM 2: Augur

Augur (Froehlich and Dourish 2004) is line oriented source code visualization. It was developed for two primary tasks; monitoring activity in a distributed software project, and exploring the distribution of activities in time and space. It consists of a number of linked visualizations.

4.4.1Implementation

Augur requires no additional setup to run; however, it uses the information already available in configuration management systems.

4.4.2Views and Interaction

(1)OVERVIEW

An overview, which is a line oriented representation for the code. Each line is colored to represent some attribute, for example the author or modification date. This view is inspired by SeeSoft (Eick, Steffen et al. 1992). Along with the primary view, there are other views adding more information about the authors of the code and the evolution of the code as in Figure 9.

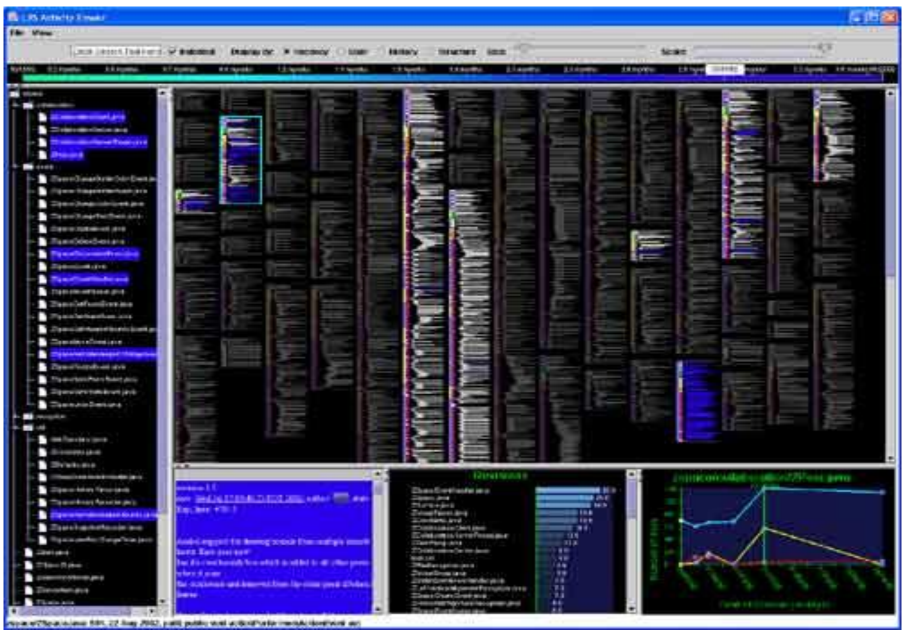

Figure 9: Augur Overview. Used with permission of author, Jon Froehlich, University of Washington, 2004.

(2)HISTORY VIEW

Figure 10 shows the history view in the Augur interface. In the history view, lines are colored according to their change history.

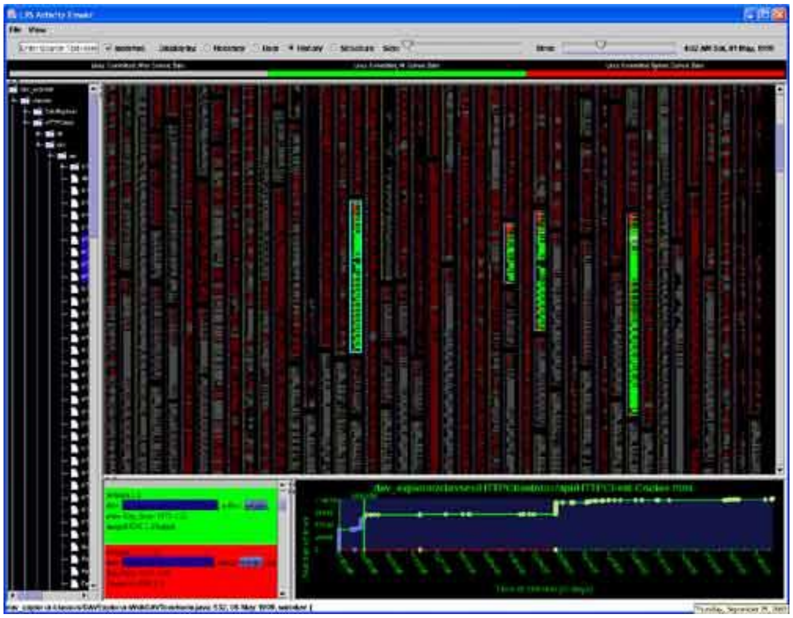

Figure 10: Augur History View. Used with permission of author, Jon Froehlich, University of Washington, 2004.

(3)AUTHOR VIEW

Similarly, the author view in Figure 11 shows the authors involved in the development of a certain file.

Figure 11: Augur Author View. Used with permission of author, Jon Froehlich, University of Washington, 2004.

Augur uses three techniques to integrate information about artifacts with its activities; annotation, interaction, and triangulation. Annotation is performed by adding some information to the basic line view of the code. Interaction is supplied by giving the user a bigger picture of some code line of interest, so if the user clicks on a certain line of code, the system automatically highlights two other sets of lines. First of these is, the lines entered at the same time of the selected line of code. Second is the structural block that contains the selected line. The third technique, triangulation, is supported by giving views in addition to the primary line oriented view. The additional views are used to give more information about some code attribute.

4.5Experiment 3

4.5.1Hypotheses

In order to work toward guidelines for designing information on secondary large displays, we need to use source code visualization in our study. We chose two different source code visualizations, VCN and Augur, based on some variation in criteria as discussed above. The purpose of this experiment is to compare these two source code visualizations with respect to the performance of users while pursuing their tasks on the given visualization.

4.5.2Experimental Design

This experiment has the same environment setup as the previous experiment. However, there is no secondary personal display. Since the goal of this experiment and the next one is to develop design guidelines for information design on large displays, we decided to use only a secondary large display in this experiment. On the primary personal display we show three code analysis questions. This experiment explores two conditions; Augur condition, in which the source code visualization used in the secondary public display is Augur, and VCN condition, in which VCN replaces Augur. One free C++ project was installed on a source control server so that each of the source code visualizations can visualize this project. Three code analysis questions are asked with a time gap of 3 minutes between every two consequent questions. The questions asked in each condition are almost the same with slight modification that doesn't change the logic of the answer. The number of participants and the nature of the conditions lead to employing a Latin Square design.

Each participant by the end of the experiment is encouraged to fill in a questionnaire about his preferences as to the conditions presented in the experiment. The whole experiment takes around 9 minutes for each participant, plus around 5 minutes to answer the preference questions at the end of the experiment.

The independent variables in this experiment are the source code visualization and the C++ source code project. The dependent variables are the time elapsed to answer each question and the number of questions answered correctly.

4.5.3Participants

Twenty four students participated in this experiment. The participants were ten teaching assistants with excellent programming skills and fourteen students who were intermediate to experienced programmers using C++. We screened users to be fluent in English so that they could easily understand the tasks required as well as the instructions of the experiment. The average age of all participants was 20.7 (23.5 for teaching assistants, 17.9 for students) ranging from 16 to 28 years of age. Students were offered extra grades for participation and teaching assistants were paid.

4.5.4Procedures

Participants were asked to fill in a background survey. Each participant is given three code analysis questions. Each question pops up on the primary display, at a rate of one every 3 minutes. The participants interact with the source code visualization to answer these questions. If the participant exceeds the time limit of 3 minutes to answer the question, the question disappears and gets considered as a wrong answer and the next question pops up on the primary display.

At the end of the experiment, each participant fills in a questionnaire about his preference as the conditions presented, the difficulty of the questions, his opinion, and preference about each of the source code visualizations.

4.5.5Results

The results have been analyzed using RM-ANOVA for within-subject design analysis. We couldn't find a main significant effect of using VCN over Augur or vice versa. A Likert scale of 1="Strongly prefer to use VCN over Augur" to 5="Strongly prefer to use

Augur over VCN" was used in the questionnaire at the end of the study. However, the preference data analysis showed a significant user preference to use VCN over Augur for code analysis questions assigned ($p < 0.001$).

We tend to explain this insignificance of results to the fact that using any of the source code visualizations on a large peripheral display is easy and acceptable task to users. Therefore, users were able to achieve the same level of performance on both systems. However, there are various factors that would explain the tendency of users to prefer the use of VCN rather than Augur. One of these factors is the visual design of VCN since it used more eye catching colors. An even more important factor to cause this preference is the additional functionalities that VCN added such as spotting on parts of code while still preserving the big picture of the source code file, fish eye view.

4.5.6Summary

In this experiment, we were able to compare two different source code visualizations on a large peripheral display. Although we couldn't find a significant difference in user performance on any of the two systems, we were able to determine a significant user preference toward the use of VCN over Augur.

This experiment helped us to select the source code visualization to be used on a secondary public display in a collaborative software development environment. This leads to the following study in which we start employing VCN into a collaborative environment hoping to come up with a set of guidelines to designing secondary tasks on large shared displays.

4.6Interviews

To frame our experiments into one picture to draw information design guidelines for peripheral large displays, we interviewed six experienced interface designers. Two of these designers are actually HCI researchers. The semi-structured interviews involved showing the environmental setup of a collaborative environment where a source code visualization is displayed on a large peripheral display and asking for comments. Discussion points included the environmental setup, usage of large peripheral displays, information, notification, and interaction design guidelines for interfaces on shared displays. All interviewees were required to fill out a background survey and sign an agreement to record the interview for further analysis.

We designed the environment with respect to the guidelines presented above regarding the position of the large display. An empty large room was used for setting up the environment. A projector was used to project a large display on the wall. In a semi-circle or oval shape, we positioned eight desks. Each of these desks had a personal computer with a personal 17" display. Each of the interviews was conducted while the interviewee was in the environment.

4.6.1Results

Initially, we asked for comments regarding the environmental setup for a collaborative software development environment where source code visualization is displayed on a large peripheral display. After analyzing the results, we found that the comments can be divided and viewed in the following points:

- *The use of large display in a collaborative environment*: two interviewees mentioned that they look at the use of large display in a collaborative environment as a way of communication and exchanging information among users. One interviewee pointed out the criticality of using a large peripheral display in a collaborative environment by saying, "Using large displays in collaborative environment, specifically in this setup, should be with lots of care because it could lead to amounts of disruption to users, since the display is clearly viewed by all users". The other three interviewees agreed with the careful use of large displays in collaborative environment. However, they mentioned that it depends on the design of the interface and the interaction techniques. One interviewee noted that the use of large displays not only provides a way of communication, but also adds a facility of group awareness however this relies on the nature of the application design on the large display.

- *The setup of the eight disks*: All interviewees noted that all users will be able to easily view the information displayed on the peripheral display. Two interviewees mentioned that the setup is suitable for viewing the information on the peripheral display. However, they added, “this setup should be carefully thought of when designing interactions with the large display”.

- *The nature of the application used on the large display in this setup*: five of the interviewees mentioned that using source code visualization is a personal task rather than a public visualization. Although that some of the views of the visualization could help the whole group, other views won’t be of any help to the whole group at once, but might be helpful to single users.

Sharing information among a group of users can be very useful if the information matters to the whole group. Two interviewees emphasized that sharing information on large displays should be studied and designed to support almost the whole group by stating, “The key concept in using a shared display is to provide shared information that could benefit the whole group or at least most of the group at some point of time”.

Source code visualizations need considerable interaction effort from the user to filter and get the demanded information from the visualization. One interviewee commented about the use of VCN in our collaborative software development environment through noting, “…VCN is easy to use, however considerable interaction needed from the user, which I think is going to be a problem when it comes to group interacting with it”. Another two interviewees agreed that one of the major challenges in designing interfaces for collaborative environments is “group interaction”, specifically with the large peripheral display. One of these two interviewees wondered about how these users are going to collaboratively and simultaneously interact with VCN on the large display.

Using large displays leads to opportunities for sharing large amounts of information. Four interviewees emphasized one of the advantages of using large display in a collaborative environment, which is the capacity for visualizing and sharing more information. Here are some of quotes, “One of the exciting features of using a large display is that you can share large amounts of information”, “using VCN on a large display can show larger number of files, and larger view of information", and "… the more information you visualize the more chances of revealing hidden information".

Multiple views, when designing interfaces on large displays for collaborative environments, is one of the most important points to consider (North and Shneiderman 2000). Three participants mentioned that the key point in VCN for the group is the use of multiple views. Their rationale was that some of these views would be of interest to some users while other views would be of interest to other users. One interviewee mentioned that using multiple views in the visualization leads to revealing more information about the nature of the data which in turn leads to better understanding of the source code at hand.

We asked the participants whether, if they had the choice, they would use this environmental setup along with VCN. Almost all participants agreed that they would use this environmental setup. However, they weren't quite sure about using VCN on the large display. Participants mentioned that they would reconsider the use of VCN. Some of their comments are ".. I would think of another application or a modification of VCN to give more value to the whole group", "VCN is nice but it doesn't help the whole group of developers much", and " .. based on my experience, developers can use VCN on a personal basis, but they need a lot more than this to share among each others". Therefore, it can be noticed that they were quite unsure about the use of VCN not because faults or problems with VCN itself, but because the nature of a collaborative software development environment might strive for more information than what is provided by source code visualization.

4.7Guidelines

In this work we intend to develop information design guidelines for public peripheral displays. We developed two experiments to explore the differing impacts of using a large secondary display on the user's primary as well as secondary tasks in terms of performance. Based on our direct observation, we noticed a few guidelines for using large peripheral displays. The results of our experiments proved our observations and guidelines. To further explore the information design guidelines for peripheral displays, we designed one more experiment to compare two of the most known source code visualizations. Interviews were conducted after choosing one of the source code visualizations to be used in a collaborative software development environment. The analysis of the interviews further supported our guidelines and gave us the further insight of more observations.

4.7.1 Guideline 1

"*Use large peripheral displays when the situation calls for high interruption*"

The importance of using large peripheral displays is even more emphasized through our experiments, since one of the impacts of using large peripheral displays is increased primary task's performance as compared to the case when a secondary personal display is used. In addition, we showed that the performance of the secondary task is also improved using the large peripheral display given appropriate design of the secondary task.

In both experiments, we explored the interruption caused by both personal and public peripheral displays. Large peripheral displays were found to be more interruptive to users than personal ones. The comprehension differences were not found to be significant between both types of peripheral displays. Therefore, designers should tend to think of using large peripheral displays when interruption is required. However, designers should be aware that the quantitative value of the interruption could be manipulated and controlled given other design guidelines for the notifications designed on large peripheral displays.

The interview was also supportive to this guideline, since our analysis of the results show that interviewees noticed the interruption that could be caused by the use of large peripheral displays in collaborative environment. Their rationale was that the large displays are easily seen and noticed by all developers, especially given our environment setup for the collaborative software development environment.

4.7.2Guideline 2

"*The use of large peripheral displays in a collaborative environment makes clear that the information is shared or available to the whole group. Hence, it promotes group knowledge*"

Many research projects were directed to sharing information among collaborative groups using large displays such as notification collage (Greenberg and Rounding 2001) and classroom bridge (Ganoe, Somervell et al. 2003). The nature of using a large peripheral display in a collaborative environment guarantees that almost all group members see the information on the large display. The setup of the environment and the position of the display are the factors that decide whether all of the team members have a clear sight of the display or only some of them. Since the peripheral display under study is a public display then at least more than one user will be able to get information from it. Hence, large peripheral displays could be used to share information among team members. The interviews conducted were supportive to this guideline through figuring out that designers tend to think of large displays in collaborative environment as a means of sharing information and communication.

4.8Conclusion

This chapter introduced a brief history of source code visualizations with emphasize on two different source code visualizations, VCN and Augur. An empirical study was designed to investigate the effects of using different source code visualizations displayed on a large shared display on user's primary and secondary task performances. Our vision of the experiment was to get insights of useful interface design guidelines. A qualitative study is then introduced, which led to two interface design guidelines.

The research in this chapter introduced two design guidelines that tackle how and when to use large shared displays in collaborative environments. The first guideline states "*Use large peripheral displays when the situation calls for high interruption*", which emphasizes the high interruption results from the use of large shared displays rather than personal secondary ones. The second guideline states "*The use of large peripheral displays in a collaborative environment makes clear that the information is shared or available to the whole group. Hence, it promotes group knowledge*", which emphasizes the fact that using a large peripheral display in a collaborative workspace promotes group knowledge and makes sure that the information is shared to all group members.

Since our focus is directed to support collaborative software development environments with the use of large shared displays, we still need to test the validity and the effects of following these guidelines in real-life collaborative software development environment. The focus of the next chapter is to introduce a system for sharing group status information among team members using a large shared display in a collaborative software development environment.

5

SuperTrack: Implementation Details and User Study

This chapter introduces SuperTrack, presents design and implementation details, provides an in-situ study, and finally states a conclusion. SuperTrack is a system for sharing information among team members in a collaborative software development environment. It is designed of three views, Time Tracking View (TTV), Project Tracking View (PTV), and Dependencies View (DV). An in-situ study in a real software development company shows that our guidelines presented in the previous chapter would hold and introduce a successful system of sharing information using large shared display in a collaborative environment, which enhances performance of the team members.

5.1 Introduction

With the increased user involvement in running multiple tasks simultaneously, comes the challenge of utilizing the user performance in multiple tasks. Based on our work, we tend to believe that using peripheral displays would guarantee performance enhancements. Specifically, we study the use of large peripheral displays in collaborative environments. We conducted a set of experiments, user studies, and interviews to develop a set of visual information design guidelines to help designers to design interfaces on large peripheral displays.

Our guidelines are believed to be valuable and of a great benefit in various real life applications and scenarios. Educational environment is one of the various environments that could be highly improved using large peripheral displays. Various applications were designed to use peripheral displays in educational environments such as Classroom BRIDGE (Ganoe, Somervell et al. 2003). In general work environments, large peripheral displays could be used to enhance communication among employees such as Notification Collage (Greenberg and Rounding 2001). In group meetings, large peripheral displays are used to simplify the process of communicating information among meeting members (Elrod, Bruce et al. 1992).

The advancement of non-traditional computing platforms leads collaborators to seek ways this new technology can augment their environment in ways that will increase effectiveness and efficiency. The area of software development, which often relies on groups of people working together on an integrated system, suffers from various problems of quality, budget, and time. While software engineering approaches study how software project failures can be addressed by focusing on software-related theories and principles (Padayachee 2002; Nienaber and Cloete 2003; Wongthongtham, Chang et al. 2004; Berntsson-Svensson and Aurum 2006; Boonzaaier and Loggerenberg 2006), our work looks at the people—specifically how they can increase knowledge about each other's activities—toward increasing communication and cooperation among the team members.

Software development environments can be seen as collaborative environments in which team members cooperate together to produce successful software. While many software engineers would consider coding to be their primary job, collaborative environments demand that they become more ambitious in undertaking multiple tasks simultaneously such as writing code while monitoring email, tracking bugs, and instant messaging with colleagues. This research effort looks to move these types of tasks off the desktop and onto non-traditional shared displays, supporting a dual-task situation in which each user works on a primary task (coding) while attending to a secondary task (tracking overall progress of the team). It is the balance between these tasks that is of interest—with the addition of secondary displays come not only the hope of increased knowledge but also the possibility of interruption to the primary task.

Biehl et al discuss several of the group programming practices in their FASTDash system, showing the need for solutions to enhance programmers awareness in a collaborative programming environments (Biehl, Czerwinski et al. 2007). This led us to explore ways of improving the software engineering team productivity, group awareness, and communication through using large shared information displays. Specifically, we analyze characteristics of the collaborative software development environment. Hence we develop and evaluate SuperTrack, a software development tracking tool. We apply various visual design guidelines for effectively conveying information to all team members. SuperTrack is designed for software development environments in which teams consist of 2 to 15 developers that use agile software process models (Cockburn 2002) to generate code rapidly such as Rational Unified Process (RUP) (Kruchten 2003).

5.2Exploring Group Awareness Needs

In many companies, software development activities and tasks suffer from problems of missing group awareness—issues in which members of teams need to know about the activities of their fellow teammates. In their study of group awareness needs in various types of software development environments, Biehl et al found that programmers tend to enhance their group awareness through finding way to learn about other's activities (Biehl, Czerwinski et al. 2007). Moreover, Biehle et al noticed that with the existing potential to enhance group awareness using large shared displays, the existing tools were not effectively designed for use on large shared displays (Biehl, Czerwinski et al. 2007).

To understand the nature of software development, we interviewed two of the experts in the field of software development and we conducted a few on-site visits. One expert worked first as a software developer and then a team leader for 5 years. The second one has been working for 10 years as a team leader and then as an IT architect. The interviews are structured with a predefined set of questions that gets finished in around 60 minutes. The interview is composed of a list of questions that target understanding the tasks involved in building a software system among team members, starting from gathering requirements all the way to delivering the final project. Both of the interviews were recorded for further analysis and exploration. Based on these interviews, a Hierarchical Task Analysis (HTA) was built, as shown in Figure 12, to provide a means of understanding the current processes and activities undertaken to develop software.

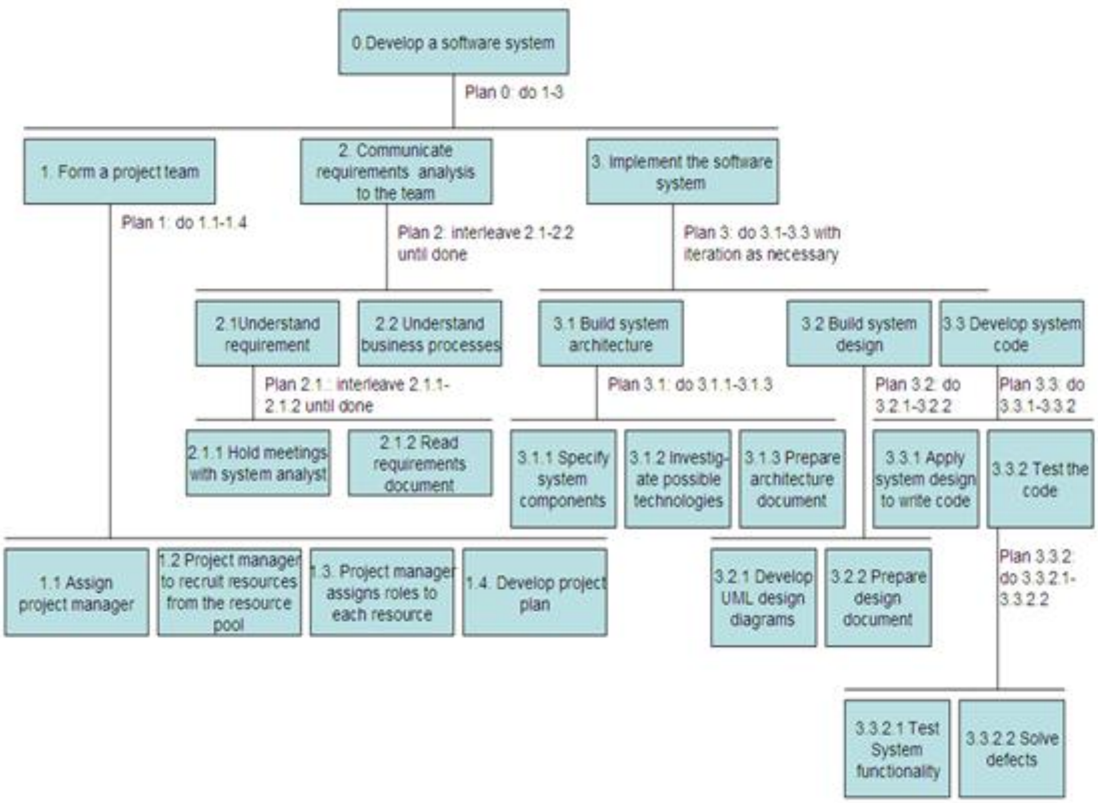

Figure 12: Hierarchal Task Analysis for the software development process followed in a big software development company

A brief study on the use of each artifact has been conducted through monitoring and interviewing the users of each artifact in an unstructured way. The artifacts studied include forms and documents used to inform team members with information. Based on the results, we summarized three major points of investigation that affect the whole team rather than individuals. We focused on information that was not necessarily of immediate and necessary importance to developers, but rather information that could help improve performance on ongoing tasks by providing enlightenment on the activities of others. These points are as follows:

1. **Sharing the status of dependencies.** Project managers spend a considerable amount of time monitoring and updating the status of external and internal dependencies. In the mean time, project managers have to report the changes of status in each dependency to all team members either through e-mails, phone calls, or even face-to-face contact in periodic meetings. Besides, team members spend time seeking the latest status of dependencies in order to make some decisions related to their tasks. In some situations, developers become idle waiting to be informed of a dependency status in order for them to resume working.

2. **Outlining task assignments.** With the increased number of projects that each team implements, the team leader gets confused regarding which tasks have been assigned to which developers. Moreover, the team leader must update the tasks and its assignments along with informing all affected team members with the new changes. Team members should know about the updates and changes to their task assignments as soon as possible in order to utilize the time and start working on the new tasks.

3. **Conveying progress status.** Project managers are mainly concerned with tracking the total progress of the project at hand and maintaining the time constraint in the project plan. Although project managers and team leaders are concerned with projects progress, team leaders are more concerned with the progress of each task separately in the project. Some automated tools exist to mitigate the problem of translating the progress of each task to a total progress. However, other team members do not feel the progress of the project since these progress reports are calculated and reported by and for team leaders and project managers. Other team members need to be aware of the status of specific tasks

and the total progress of the projects they are involved in, because this will convey the severity of the tight project deadlines to all team members.

These points highlight the design aspects necessary in a system to support activity awareness within a team. In our development effort, we assume the system will be used by team leaders and project managers co-located with the developers and other stakeholders in proximity of a large display. Each developer has a personal computer provided with a source control system to view the history of project files. The room setting is designed in a way, based on the guidelines presented in (Su and Bailey 2005), to allow all developers to easily steal a look at the large display while they are still working on their primary personal displays.

5.3Designing to Enhance Group Awareness

A clear vision for the current system of processes and activities was developed using the HTAs. Three investigation points were of interest to tackle and enhance the problem of enhancing group awareness through the use of large peripheral display. Conceptually, we designed a visual view for each of the investigation points on the large peripheral display. We hypothesize that using these views on a large shared peripheral display would enhance group awareness and hence enhance the group productivity.

In designing the views of the system, we aimed to support perception and interpretability of information. Perceiving information is concerned with creating a design where users can clearly distinguish structures in an information display. Interpreting information is defined as determining what the display elements mean and how those elements fit into the context of the interface. Notification concerns were dealt with through following our developed guidelines to provide appropriate interruption, reaction, and comprehension components.

In the following, we present the three different views along with a detailed explanation of the purpose and the functionality of each view. A discussion of design decisions undertaken for each view is also presented.

5.3.1 Time Tracking View

Project managers and team leaders are supposed to utilize the use of resources during the project lifetime in order to maximize the benefit and minimize the time required to finish the project. Human resources are considered to be one of the most important resources that a project manager is concerned with. If a developer is waiting for an internal or external dependency to resume his work, then this developer needs to be quickly spotted and given another task that doesn't wait for a dependency. Following this strategy would lead to a maximum utilization for this developer and hence higher productivity in terms of project activities accomplished.

The *time tracking view* (TTV), Figure 13, does just this. Specifically, TTV simplifies the process of quickly spotting the status of human resources along with activities. To gauge the progress of specific activity, the human resources engaged in this activity need to be known. TTV shows all members involved in specific activity along with the progress in percentage of the accomplishment that each member finished in his assigned task of the activity. Moreover, TTV is designed to simplify the process of spotting free resources, resources that aren't assigned to activities, activities that aren't assigned resources, finished activities, and members that are currently free to get assigned other activities and tasks.

The first design decision concerning TTV is the visualization structure for activities along with members. Activities and members are two factors of concern that have a matrix like combination among each others. Therefore, the well known x-y chart presentation is used for TTV. The x axis presents the different project activities obtained from the break down structure of the software at hand during the planning and analysis phases, while the y axis presents the members currently participating in the software project development.

In each intersection between an activity and a member, we need to show two types of information. (1) The assignment of the member to the activity. (2) The progress of the member's assigned tasks in the activity. Therefore, a progress bar found to be the most appropriate representation for the progress of each member assigned to this activity. However, only the progress bar doesn't convey the information of whether the member is assigned a specific activity or not. Therefore, we decided to show a disabled form of a progress bar to represent a member that isn't assigned to a specific activity. Since progress bar doesn't really indicate the accurate percentage, we decided to show the exact percentage in the center of each progress bar which facilitates gaining more accurate information.

Looking at Figure 13 could easily convey meaningful information to the team leader and project manager such as the performance of each individual team member during the project. Hence, the team leader can categorize high, medium, and low performance members among the team. For example, team member M1 finished 70% of activity A1. Users can easily notice that member m2 isn't assigned to activity A1. Moreover, member M3 is assigned to activities A1, A2, A3, and A4. However, M3 has no progress at all concerning activities A1 and A3.

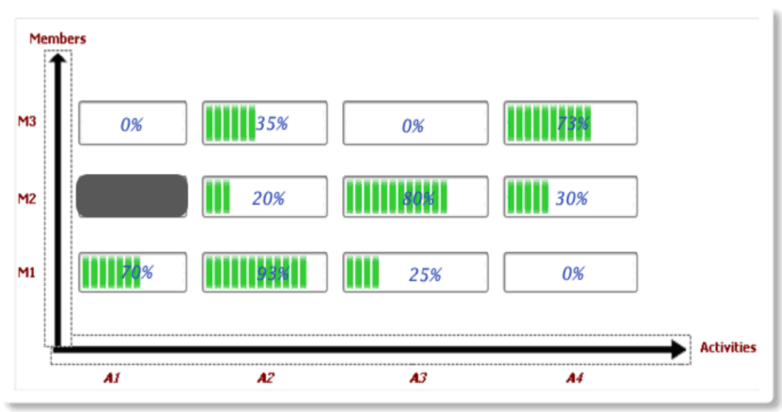

Figure 13: Time tracking view showing that member M2 is not assigned activity A1

Since effectively managing resources is crucial for the success of software projects, this view tends to notify the project manager along with the other team members when a resource, such as a team member, is free and ready to be used in other activities. The notification needs to be interruptive so that the project manager can easily notice and quickly start taking action. Therefore, we needed a notification about a resource on the X row that either finished all tasks in all assigned activities or isn't assigned any tasks in any activities. Hence, we decided to make the X row, representing this resource, flash for

30 seconds in a relatively slow rate of one flash every second. For those users who were involved in their primary task and couldn't notice the notification, we agreed that the whole row gets colored in light grey if the resource is free such as in Figure 14.

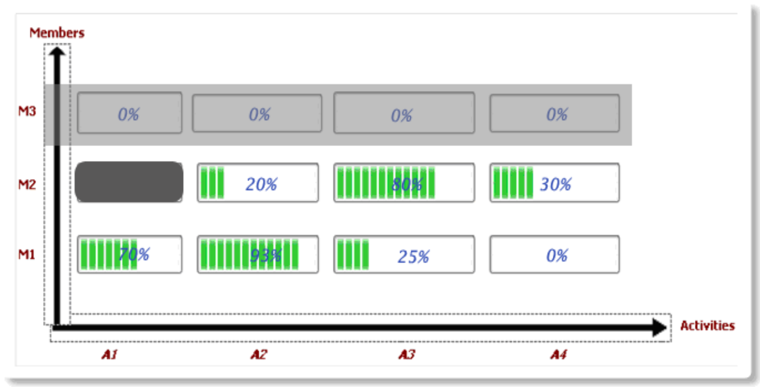

Figure 14: Time tracking view showing that member M3 finished all his work and is considered an empty resource that should be utilized

5.3.2Project Tracking View

Project managers need to keep a track of the project progress in order to generate other status reports, including current actual cost as compared to estimation cost, the general progress of the project, and further requirements and scope modification documents. Team leaders are required to report the project progress to higher management including the project manager. Sharing the project progress can have its impact among all team members. For example, developers who are currently delayed in a specific task can start seeking for more help or notify the team leader since tight deadlines are in front of them. Moreover, team leaders need to be aware of the progress of each activity along with the total project progress. Preferably, if team leader can get more information about specific activity such as who are the team members assigned this specific activity. Getting this detailed piece of information would help team leaders to spot the activity or activities causing the delay, hence take action.

The *project tracking view* (PTV) is designed to facilitate sharing the previously mentioned functions among team members. In more details, PTV shows the existing activities that are currently running in the project along with its current progress. In addition, PTV shows the total progress of the project. Hence, team leaders can easily, from one glance at the large shared peripheral display, be aware of the total progress of the project they are involved in. The team members assigned for each activity are also visualized in PTV to simplify the process of progress analysis in terms of delays causes or resources management.

As for the design of PTV, it is noticed that we need to visualize progress of activities along with a total progress. Therefore, we tend to use a progress bar to demonstrate or represent the progress of each activity separately along with the total progress as shown in Figure 15. Like TTV, the use of progress bars doesn’t convey accurate information about the current progress. Therefore, we show the accurate percentage in text centered in the progress bar. Each activity is represented in a rectangle or a box with the activity’s title on top. Below the title, a list of names of team members assigned to this activity is displayed. The last thing in the box is the progress bar that shows the activity of this specific activity. The total project progress is showed underneath all activities and the progress bar itself is a little wider than the other progress bars to represent that it is a total project progress, not a single activity progress. This thickness in the total progress bar acts as a sort of emphasis on this specific information.

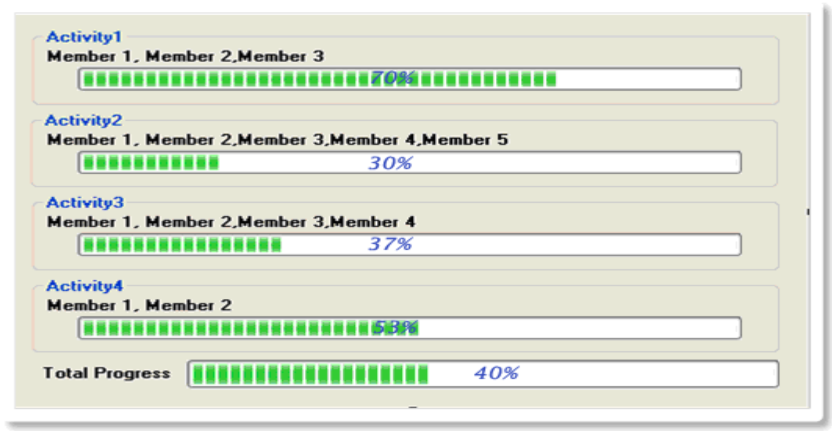

Figure 15: Project tracking view showing the progress of each activity along with the members assigned to work in each activity

When a specific activity gets fully accomplished, its box flashes for 30 seconds similar to the flash mentioned in TTV and then disappear. This in turn continuously frees more space in PTV to show other new activities.

The combination of TTV with PTV can be very beneficial to team leader. For example, team leader can look at PTV to get information about the progress of each activity along with the total progress of the project. In case that the team leader notices an activity that he believes is delayed or slowly advances, he can know whom is assigned to finish this activity. Assuming that there are 4 team members are assigned to this activity, team leader can look at TTV to know the progress that each of these team members achieved in the tasks assigned to him for this activity. Hence, team leader can specifically and quickly spot the human resource that is facing a problem and can start communicating with this team member to speed up the process.

5.3.3Dependencies View

Not only team leaders, but also all members of the project team are concerned with the status of dependencies throughout the project life cycle. These dependencies can substantially affect the progress of the entire team if delayed. Therefore, it is crucial that dependencies status gets shared among team members as soon as it gets changed so that the concerned team starts taking appropriate actions. There are two types of dependencies. The first type is an external dependency which is defined as an extrinsic factor that should be available before starting the dependent project activity. The second type of dependencies is an internal dependency which is focuses on the logical and technical relation among the various activities in the project.

The *dependencies view* (DV) is intended to show the status of the dependencies along with its classification whether an external or internal dependency. Moreover, it provides a calendar of the expected dates for specific dependencies to be finished. Moreover, the duration of the dependency is also conveyed in DV. The title of each dependency is provided along with a brief description of the dependency. This view is considered to be valuable among all team members due to saving communication time needed to share the information of dependencies among all team members.

Each of the dependencies whether internal or external has duration, start date, and end date for the activity to be finished at as shown in Figure 16. Moreover, some dependencies depend on each others such as internal dependencies depending on each other or an internal dependency depending on an external one. Therefore, a Gantt-Chart like representation is chosen. The rationale behind choosing Gantt-Chart like representation is because all team members are already familiar with the representation and interpretation of Gantt-Charts which would make it easier for them to understand and learn the view. We color coded the type of dependency with a red color for external dependencies and black for internal ones. Colors were also used to convey the information of an accomplished dependency. A light gray is used to represent a finished dependency.

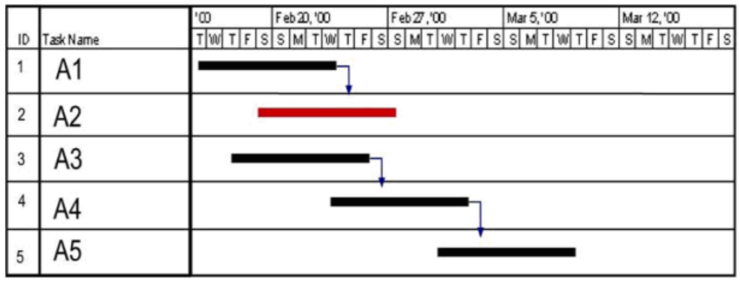

Figure 16: Dependencies view showing the dependencies of the project, discriminating internal dependencies with black color, and external dependencies with red color

The three views discussed above are arranged in a large shared peripheral display as in Figure 17. The coordination of colors and labels among the views was designed to allow users to quickly obtain and associate the information most important to them. The consistent positioning of each view allows users to focus on the view (or views) most important to them, as we assumed that different users would have different levels of interest in each view. Use of motion and animation on the views that might not be tolerable on a desktop display should be more welcomed by users when the display is not in their immediate proximity. While there are certainly many more views and alternate layouts of the views that could be considered, the variety and layout of views shown in Figure 17 should provide insight in an in-situ study as to possible benefits for coordinated group information on a shared display.

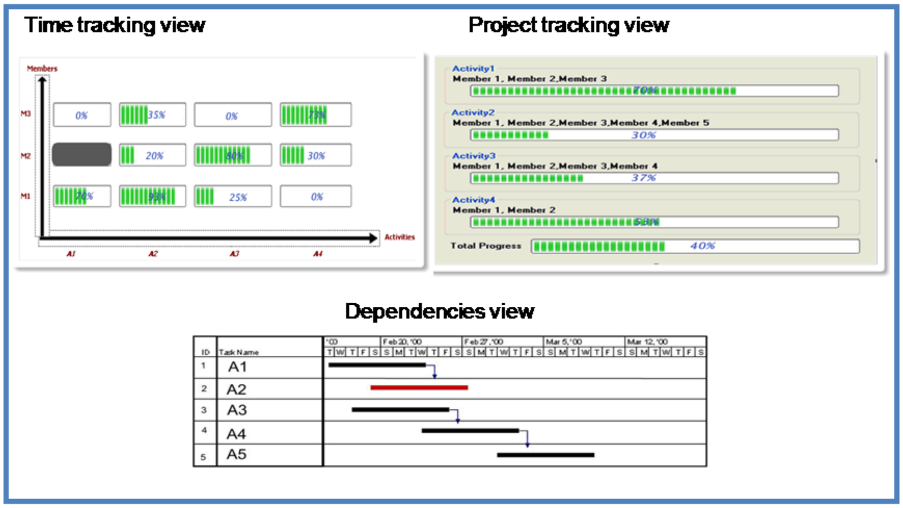

Figure 17: SuperTrack overview with its three views

5.4 In-Situ Study

To explore the utility of SuperTrack, in particular its effectiveness with respect to our metrics (possible performance gain and interruption costs resulting from additional information about collaborative partners), we deployed SuperTrack in a medium-sized software team for 10 weeks. We interviewed and surveyed managers and developers to understand how they used and would use a tool like SuperTrack. While conclusive and statistically-significant findings are difficult to make and assess from such a study, the comments from the users support many of our hypotheses and point us to new and interesting areas for future study.

5.4.1 Hypotheses

In this work, we have a focus on how a large shared information display deployed in a collaborative environment can improve the performance of team members. Specifically, the goal of the study is to explore how SuperTrack, with its three views, in a software project of medium size (2-15 members) can help improving the performance of the team due to increased awareness of team members. Moreover, we hypothesize that increasing the group awareness among team members leads to increased communication, hence maximizing the utilization of each team member in terms of finishing tasks in less time than planned.

5.4.2 Study Overview

To check the validity of our hypotheses, we developed an in-situ study in a mid-sized software development company. The study is designed to explore the performance change associated with each team member and the amount of saved time in the activities of a specific software project. Moreover, the study involves a set of interviews. Various stakeholders were required to do the semi-structured interviews. Specifically, the interviews engaged project manager, team leader, senior and junior developers. The interviews focus on two dimensions. The first dimension is concerned with the value of sharing software project information on a large shared peripheral display in a collaborative software development environment. The second dimension tackles the value of the information being shared in terms of collaboration, group awareness, and team productivity or performance. The interviews were noted to take 60 minutes in average for each interview. All interviews were recorded so that we can consolidate the answers in order to form concrete opinions about the major concepts, including the effects of using large peripheral display in collaborative environments and information design guidelines for large peripheral displays.

5.4.3 Participants

Twelve participants volunteered in our study, forming a comprehensive team. The team included project manager, team leader and ten software developers as in Table 2. The project manager has come from a technical background since he used to be software developer for 7 years and directed his career towards project management 5 years before this study. The team leader is technical person with leadership skills that qualified him to occupy this position 3 years prior to this study after working as a software developer for 4 years. The developers are all with good JAVA programming skills. Their skill level is gauged using a skill assessment scheme specified as part of the software process used in the software development company hosting the study. Six of developers are junior developers with skill level 3 on a 5 skill level scale in which 5 is an expert. The other four are senior JAVA developers with skill level 4 and 5. We screened participants of average age of 28.4 ranging from 23 to 35 years of age. Employees participated in the experiment were given a small monetary reward for their time and efforts.

Table 2: Participants roles in the in-Situ study

Number of participants	Role of participant
1	Project manager
1	Team leader
6	Junior JAVA developers
4	Senior JAVA developers

5.4.4 Materials

A mid-sized software project is selected to be the software project under study. The project was in its early stages—only project plan was developed without considering the use of SuperTrack. The software company hosting the study provided us with a well equipped environment to run our study. The environment consists of a single room that is suitable to have all team members working simultaneously. The room contains twelve desktop PCs assigned to the twelve participants. The twelve participants were seated in a semi-rectangle with one open side, where the large screen display is deployed as illustrated in Figure 18. Again we followed the environment setup mentioned in the guidelines for placing large displays (Su and Bailey 2005).

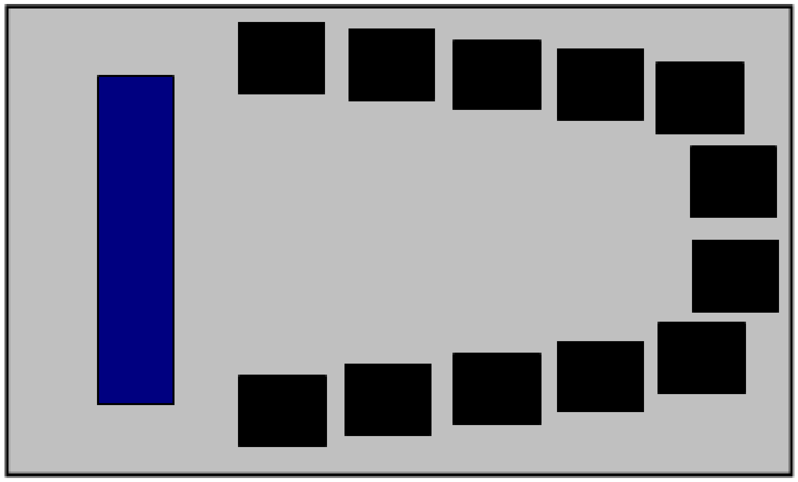

Figure 18: The room arrangement of the experiment

5.4.5 Procedure

A training session was presented to all participants introducing SuperTrack. Following the session, a digested manual on how SuperTrack works was provided to each participant. All participants are asked to track their working hours on a time tracking sheet that we designed to show how each participant has spent his time. We asked participants to fill in a column in the time tracking sheet showing whether the time spent is planned or an ad-hoc activity. Using this time tracking sheet, we can exclude those tasks that are not in the original plan of the project as we are interested in investigating the performance gain for each participant when compared to the planned durations of each task.

5.4.6Results and Discussion

Observations

In the beginning, it is noticed that developers were highly interrupted and dissatisfied with the introduction of SuperTrack and the use of larger peripheral display in their work environment. Slowly, this disruption and dissatisfaction quickly diminished with time before roughly finishing the first quarter of estimated project time. On the other hand, we noticed seamless interaction and appreciation of SuperTrack along with the shared display from the project manager and the team leader. As for the environment setup and the arrangement of seating, all team members seemed accepting it and easily viewing the information presented on the shared display.

Based on the analysis of collected time sheets, we noticed that project manager and team leader were able to complete tasks, such as monitoring project progress, filling resources usage reports, and tracking project dependencies status, roughly 15% faster compared to expected performance. Although the estimated time plans were professionally designed based on previous experiences, this doesn't guarantee the accurate time estimations to provide real scientific proofs. However, this shows promising increase in productivity through saving time at some tasks for project managers and team leaders. One way to explain the productivity increase for project managers and team leaders is the existence of views in SuperTrack that automate some tasks that were used to be done manually such as notifying team members about the current status of a specific dependency. Moreover, it is noticed that some project manager and team leader tasks were diminished through increasing the group awareness through the shared information. Regarding software developers, a relatively slow performance increase approaching 7% was noticed in their development task. Therefore, we believe that there is something that needs further investigation regarding the use of shared information displays for software developers.

These notices and the results we got from our analysis of estimated and actually times are further investigated and discussed in the following section. Interviews were conducted to further get more insight of the nature of performance gains and deeper understanding of how the tasks got modified to fit the new environment.

Interviews

Initially all participants were asked about their opinion about the use of shared information displays in their work environment. The project manager and the team leader were very enthusiastic in the way they answered this question. Their replies express high satisfaction about the experience of having a shared information display among the group. The project manager stated, "*Since it is a shared display, using it for sharing information among my team sounds very appealing to me*". Moreover, one of the team leader comments is "*… team work strives for sharing information, which is accomplished using the peripheral display*". The team leader added another comment, "*…knowing the progress of my team from one glance at the large display makes my life easier and it helps me to have tangible evidence to motivate them*".

As for developers, the entire group conveyed their dissatisfaction to the introduction of shared information display in the beginning of the project. However, they mentioned that this dissatisfaction turned into appealing when the project got into more tense state. For example, one developer noted, " *… in the beginning of the project, me and my mates were like very annoyed with the whole experiment since the display looked annoying to us because we aren't used to it, and we were unsure of what benefits we can get out of it. However, when the project got tense, personally, I found very important benefits from sharing information about dependencies*". Another developed stated, "*…sharing information sounds to me an important thing. However, I don't see that SuperTrack was the perfect information sharing tool for us*". We can see from the preliminary questions that the information displayed along with they way it is presented in SuperTrack needs more analysis.

In general the largest single segment of the team—the developers--were reluctant to change of the system in the beginning, due to the high interruption factor they mentioned and the unseen benefit from time tracking and project tracking view for them. Perhaps in no small part they felt upset at the display of their own "private" information—which had always been available to others but had not previously been displayed so prominently. We believe that they might have felt that there is too much information in the large display, and they get benefits from only a small portion of the views. Much as the students in our classroom study gained little information from the large display in their work area (Ganoe, Somervell et al. 2003), the workers did not at first understand the benefits of the additional information. However, it was encouraging to see that over time they began to appreciate its benefits.

Aside from all the other guidelines to when to use multiple views when designing information visualizations (Baldonado, Woodruff et al. 2000), it is recommended to use multiple views when a group with different interests is targeted with the visualization. The project manager emphasized this through stating, “*Of course, the nature of information needed by me is different than what would be interesting for developers. Therefore, I think that using multiple views is a good idea*”. On the hand, one of the developers commented, “*Using multiple views means providing specific information for specific individuals. This means losing the value of sharing information*”. The last opinion is correct, however, in any team, there is a subgroup that is interested in the same information. Therefore, we believe that using multiple views in shared information displays in collaborative environments is crucial to communicating more relevant information.

Generally, SuperTrack was helpful to project manager and team leader more than developers. When we asked the team about their general opinion about the specific use of SuperTrack, project manager and team leader were very excited about it but, this wasn’t the case with developers. The project manager noted, “*…SuperTrack is very interesting for me, however, I don’t know how it would be useful for the rest of the team*”.

One of the goals behind designing SuperTrack is to be an application of our shared information design guidelines. Therefore, we started to direct our interview towards the design guidelines of SuperTrack. Some questions tackled the design and the presentation of information in the TTV. The project manager commented, “*TTV is very good view for me to get more quick information about the human resources at my team*”. The interruption caused by the flashing of a free resource was helpful and didn’t cause any dissatisfaction. The team leader mentioned that the flashing is a good indicator for him that there is an action that needs to be taken regarding this member. As for developers, they mentioned that the flash was interruptive to them, but thirty seconds didn’t cause them any harm or any degradation to their primary task. One developer noted, “*Although my team mates don’t like this view, for me since I am a junior developer, when I see the flashing of a senior developer, I run and ask for his help if I am stuck in something*”.

After all, TTV seems to be of an indirect benefit to developers as of a direct benefit to team leader and project manager.

PTV seems to be of no benefit to developers since the majority of the team answered the interview questions regarding this view in an unpleasant way. One developer mentioned it all, “*Personally, I don’t see a benefit of this view for me. I talked with my friends in the team and it looks like none of us found a benefit in it*”. Unlike developers, team leader was very pleasant with PTV since it helps him in quickly filling detailed paper reports of the project progress. However, the team leader emphasized the developer’s point through stating, “*… although this view is helpful for me as I told you, I don’t see any usefulness to my team from it*”. The interruption caused by the flashing of a finished activity was a nice (cool) notification for team leader. The team leader noted, “*The flashing of a finished activity for 30 seconds, and then the activity disappears was cool, I liked it*”.

The DV provides an easy way of sharing information about the status of internal (finishing implementation of a module) and external dependencies (buying a component). Developers were highly excited and enthusiastic regarding DV. Some of the developers mentioned that DV was the only view they were interested in. One of the developers stated, “*SuperTrack wasn’t helpful for me with the exception of the dependencies view. This view was drastically saving me time and considerable effort*”. Developers explained that the old way of getting the information about the dependencies status was to keep trying to get in contact with the project manager and/or the team leader, which was time consuming and not productive.

In terms of improving the experience of developers, they wished if SuperTrack provides facilities of saving information and hence, they could print time sheets and save time writing and filling these time sheets for project managers. One of the developers mentioned, “*SuperTrack could be VERY helpful for us if it have archiving facilities to save us time write the same information presented on the display*”.

5.4.7Summary

In this study, we introduced SuperTrack, a system for sharing information in a collaborative software development environment, in a real-life software development company. The team used SuperTrack throughout the entire life cycle of the project. Our analysis of the interviews shows promising opportunities to using large shared displays in collaborative software development environments. The design guidelines, we developed earlier, proved to be of applicability since the team members commented positively regarding the interruption and the share of information among the team members which promotes group awareness. The group awareness increased due to sharing information about the status of each team member and the activities they are involved in. Moreover, the performance of the team is improved and the project was finished in less time than what was planned by the project manager. Despite the success of introducing SuperTrack, we noticed that the information displayed needs to be further analyzed since developers mostly were only interested in one view of the three.

5.5Conclusion

This chapter presented a new system for sharing information in a collaborative software development environment using a large shared display. The implementation of this system followed the design guidelines we presented in the previous chapter. SuperTrack is composed of three views; Time Tracking View (TTV), Project Tracking View (PTV), and Dependencies View (DV). The chapter provided an in-situ study to study the performance differences of the team members and the effects of introducing SuperTrack with a large shared display in a real life collaborative software development environment. The results of the study show promising results with respect to the design guidelines presented in chapter 4. Moreover, team members emphasized the awareness increase from the use of the large shared display and SuperTrack, which resulted in increasing the team performance. Although, the information presented in the views were accurately designed, it still needs more analysis since developers were only interested in one view, where we believe that there are other information that would further enhance the developers' performance.

6

Conclusion and Future Work

This chapter concludes the most important contributions presented in this thesis. A brief description of each contribution is also provided with consideration of linking findings and contributions to the big picture. Finally, future work is presented.

6.1Summary of the Work

This thesis focused on understanding the effects of introducing large shared displays in collaborative software development environments. An initial study was intended to explore the different performance effects between using a secondary personal and public displays in terms of primary and secondary tasks performance. The study shows significant performance improvements of the secondary tasks with the use or large shared display rather than secondary personal one. Although the primary task performance degraded with the introduction of secondary displays, the performance is better when the large shared display is used as opposed to the secondary personal one. Another experimental study was presented with the intention to further study the different design features for two different source code visualizations, VCN and Augur. Results show that VCN was preferred by users due to its use of overall cushions and other features that made it easier for participants to accomplish their task. A qualitative study was then accomplished through interviewing participants with expertise in interface design. This study was designed to analyze and assess interface design guidelines for large shared displays in collaborative environments. Two guidelines were developed, where the first one tackles the interruption component of the large shared display to the group and the second one tackles the fact of sharing information among group members and promoting collaboration and group awareness. SuperTrack was implemented as a proof of concept to employ the guidelines we developed. SuperTrack is an information sharing application for collaborative software development environments. An assessment of SuperTrack was developed in a software development company, leading to further insights into analyzing the data being viewed using SuperTrack and proving the benefits of using the interface design guidelines and using large shared display in the software development environments. Team performance is shown to be improved and the group awareness along with the collaboration has been increased among team members.

6.2Major Findings and Contributions

This thesis aimed at understanding the effects of introducing large shared displays in collaborative environments. The major findings are the following:

- *Large shared displays are more interruptive and provide higher comprehension than secondary personal ones*: Results from Chapter 3 showed that the large shared displays are more interruptive to users than personal ones, since participants were able to more notice the changes in the secondary task than when personal display was used. Measuring the comprehension of the secondary task in both experiments in Chapter 3 showed that there is a higher comprehension component associated with the use of large shared displays. Chapter 4 introduced a qualitative study with the purpose of designing guidelines for large shared displays. The results of the qualitative results further support that large shared displays are more interruptive and provide higher comprehension than secondary personal ones. Chapter 5 further supports the findings through building a system that uses a large shared display in collaborative software development environment.

- *The use of large shared display enhances the user performance on the primary task as compared to the use of secondary personal displays*: Based on the results of the experiments in Chapter 3 along with the explanation of finding a higher interruption component with the use of large shared displays, we found a higher primary task performance when a large shared display is used rather than personal one. Chapter 5 in-Situ study results supported this finding through assessing the primary task performance of the project manager, team leader, and developers.

- *The use of large shared display significantly improves the user performance on the secondary task*: Similarly, results from Chapter 3 showed that there is significant performance improvement when a large shared display is used rather than personal one. These results were duplicated and proved in two different experiments, where one of them used lab designed tasks while the other experiment used real life primary and secondary tasks.

The major contributions are the following:

- *Two design guidelines for using large shared displays in collaborative environments*: In this thesis, we aimed to understand the different effects of using large shared displays on the user's tasks. Based on the findings we summarized, we started thinking of user interface guidelines for large shared displays with the goal of improving the user performance. Our qualitative results in Chapter 4 helped us in developing two interface design guidelines for using large shared displays in collaborative environments. The first guideline states using large shared displays when the situation calls for high interruption, while the second guideline states that using large shared displays should be used to promote collaboration, group awareness, and ensures sharing information among group members.

- *A system that acts as a proof of applicability of the design guidelines for large shared displays in collaborative software development environments*: In this thesis, we had a focus on improving communication and group awareness in collaborative software development environments. The findings in Chapter 3 helped in designing user interface guidelines for large shared displays. To fully understand the effects of large shared displays and to assess our developed user interface guidelines, we developed a system for sharing information among group members in a collaborative software development environment in Chapter 5. The system used the previously mentioned design guidelines and showed performance improvements of team members, enhanced collaboration, and increased group awareness.

These conclusions confirm that there is a different impact of using personal secondary display as opposed to large shared displays. Large shared displays not only more interruptive to users than secondary personal displays but also enables users to have higher comprehension compared to what they get from using a secondary personal display. In addition, the conclusions provide evidence that the two design guidelines presented hold true when applied on systems that use large shared displays in collaborative software development environments.

6.3Future Work

In this thesis, we isolated the effect of interaction when studying the effects of public and personal displays on user's primary and secondary tasks performance. Therefore, we believe that understanding the differences between using secondary personal displays as compared to large shared displays needs to be further studied through a set of experiments that would target the different interaction techniques accessible for each display. Some research has been directed to these types of effects such as Karam's work (Karam 2006) however, we believe that enhancements to this work and relating it to the IRC framework would lead to promising insights. Furthermore, our results as well as other effects need to be investigated if the large shared display is replaced by pixel-dense large displays.

In chapter 4, we developed two design guidelines. However, we believe that a large set of other guidelines need to be further explored. Other guidelines for using interfaces on shared displays would include guidelines for interaction techniques, guidelines for using multiple views, and guidelines for sharing information for different interest groups.

Although SuperTrack was successful in sharing information among group members in the collaborative software development under study, it still needs to further be analyzed and enhanced. The number of views, the information presented in each view, and other information that could be shared to enhance the group performance are all enhancements that could be added to SuperTrack to increase the productivity of the group. Moreover, SuperTrack needs to be fed with versatile interaction techniques to further enhance the group interactivity, cooperation, and interactivity with each others.

Appendices

Appendix A

First Round

1. What was the color of the tuxedo?

   - ○ Yellow
   - ○ White
   - ○ Black

2. How many lions were there?

   - ○ 1
   - ○ 2
   - ○ 3

3. How many tennis balls were there?

   - ○ 1
   - ○ 2
   - ○ 3

4. What's the color of the taller cup?

   - ○ Yellow
   - ○ Blue
   - ○ Brown

5. Was there any food on the table?

- Yes
- No

6. What was the color of the fan?

- Green
- Yellow
- Blue

7. What was the color of the phone?

- White
- Black
- Green

8. What was the color of the smaller dog?

- Brown
- Black
- White

9. What was the fruit?

- Apple
- Orange
- Watermelon

10. How many martial artists were there?

- 1

- 2
- 3

11. What was the name of the café?

- Cilantro
- Carlos
- Costa

12. How many laptops were there?

- 1
- 2
- 3

Second Round

1. What was the color of the wrist watch?

- Blue
- White
- Black

2. What was the color of the flower?

- Blue
- Pink
- Yellow

3. Was the baby wearing a hat?

   - Yes
   - No

4. What was the color of sandal?

   - Yellow
   - Blue
   - Brown

5. What was the color of the book?

   - Blue
   - Green
   - Brown

6. What was the color of the smaller TV?

   - Black
   - Brown
   - Blue

7. What was the color of the car?

   - Silver
   - Black
   - Green

8. How many monitors are there?

- 1
- 2
- 3

9. What was the color of the virtual keyboard?

- Green
- Black
- Red

10. How many cameras were there?

- 1
- 2
- 3

11. What was the color of the sun glass?

- Red
- Black
- Yellow

12. How many knives were there?

- 1
- 2
- 3

Appendix B

First experiment results

Participants	Base (WP3Mins)	Personal Display (WP3Mins)	Large Display (WP3Mins)	Personal Display (Questions)	Large Display (Questions)
1	157	145	140	11	11
2	210	198	195	11	12
3	64	57	55	9	10
4	84	79	72	8	10
5	200	186	183	10	10
6	115	90	100	10	12
7	139	130	125	10	10
8	82	69	67	11	12
9	70	50	55	12	11
10	138	126	124	10	10
11	108	94	92	12	10
12	224	219	200	9	11
13	136	129	123	10	12
14	155	143	140	8	12

15	221	211	209	12	11
16	171	165	154	9	11
17	120	112	98	9	12
18	107	96	95	11	12
19	214	162	168	12	10
20	231	195	198	12	12
21	109	85	83	11	12
22	206	191	189	9	11
23	142	125	120	12	10
24	137	132	124	8	10
25	191	187	180	9	12
26	153	138	132	10	12
27	73	60	54	11	10
28	196	182	176	10	12
29	65	56	50	9	11
30	93	80	75	11	12

Second experiment results

Participants	Base Condition (Time Sec)	Personal (Primary Time Sec)	Public (Primary Time Sec)	Personal Questions (Time Sec)	Public Questions (Time Sec)
1	11.59	10.79	11.90	2.23	1.65
2	10.78	11.75	10.37	2.54	2.03
3	10.17	11.87	11.11	3.20	2.36
4	8.94	11.29	10.16	1.98	1.96
5	9.02	11.08	9.17	2.54	2.95
6	11.85	9.31	10.00	3.14	2.49
7	9.13	10.89	8.20	2.23	2.06
8	9.85	9.41	9.28	2.53	2.58
9	11.28	11.03	8.98	2.28	2.44
10	10.63	10.69	9.97	2.46	1.99
11	10.33	10.44	11.39	2.17	2.41
12	11.14	10.16	9.67	2.56	1.68
13	8.49	8.82	8.26	2.26	2.55
14	9.84	9.08	10.08	2.27	1.68
15	10.34	10.20	9.70	3.46	2.08
16	9.36	9.35	11.22	1.98	1.58

17	8.27	11.57	10.37	1.66	1.32
18	10.92	11.38	8.68	1.78	2.23
19	9.17	11.53	10.52	2.38	2.28
20	9.67	9.27	11.16	2.76	2.22
21	10.59	10.78	9.50	1.97	1.24
22	9.09	11.18	11.47	1.84	1.71
23	10.40	11.63	11.24	2.52	2.40
24	8.39	11.70	8.68	2.41	1.72
25	9.30	11.52	8.36	2.69	1.70
26	10.65	8.09	11.61	1.90	2.14
27	10.88	11.84	9.05	2.75	1.52
28	11.99	10.53	11.09	2.64	2.20
29	8.00	8.88	8.97	2.18	1.99
30	9.94	8.59	9.72	3.17	1.22

REFERENCES

Anderson, J. A., J. Colvin, et al. (2003). Productivity and multi-screen displays, Technical Report by Lindsay Research & Consultin, Inc.

Baldonado, M. Q. W., A. Woodruff, et al. (2000). Guidelines for using multiple views in information visualization. Proceedings of the working conference on Advanced visual interfaces. Palermo, Italy, ACM Press.

Berntsson-Svensson, R. and A. Aurum (2006). Successful software project and products: An empirical investigation. Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering. Rio de Janeiro, Brazil, ACM Press.

Biehl, J. T., M. Czerwinski, et al. (2007). FASTDash: a visual dashboard for fostering awareness in software teams. Proceedings of the SIGCHI conference on Human factors in computing systems. San Jose, California, USA, ACM Press.

Boonzaaier, J. and J. J. V. Loggerenberg (2006). Implementation of a project office to improve on project delivery and performance: a case study. Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing couuntries. Somerset West, South Africa, South African Institute for Computer Scientists and Information Technologists.

Booth, K. S., B. D. Fisher, et al. (2002). The "mighty mouse" multi-screen collaboration tool. Proceedings of the 15th annual ACM symposium on User interface software and technology. Paris, France, ACM Press.

Chewar, C. M. (2005). User-Centered Critical Parameters for Design Specification, Evaluation, and Reuse: Modeling Goals and Effects of Notification Systems. Computer Science and Applications. Blacksburg, VA, Virginia Polytechnic Institute and State University. **Doctor of Philosophy**.

Chewar, C. M., E. Bachetti, et al. (2004). Automating a design reuse facility with critical parameters: Lessons learned in developing the LINK-UP systems. Computer-Aided Design of User Interfaces (CADUI).

Chewar, C. M. and D. S. McCrickard (2003). Educating Novice Developers of Notification Systems: Targeting User-Goals with a Conceptual Framework. World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003, Honolulu, Hawaii, USA, AACE.

Chewar, C. M., D. S. McCrickard, et al. (2004). Unpacking critical parameters for interface design: evaluating notification systems with the IRC framework. Proceedings of the 2004 conference on Designing interactive systems: processes, practices, methods, and techniques. Cambridge, MA, USA, ACM Press.

Cockburn, A. (2002). Agile Software Development, Addison-Wesley Professional.

Colvin, J., N. Tobler, et al. (2004). "Productivity and Multi-Screen Displays." Rocky Mountain Communication **2**(1): 31-35.

Czerwinski, M., G. Smith, et al. (2003). Toward Characterizing the Productivity Benefits of Very Large Displays. INTERACT**:** 9-16.

Dudfield, H. J., C. Macklin, et al. (2001). Big is better? Human factors issues of large screen displays with military command teams.

Eick, S. G., J. L. Steffen, et al. (1992). "Seesoft-A Tool for Visualizing Line Oriented Software Statistics." IEEE Trans. Softw. Eng. **18**(11): 957-968.

Elrod, S., R. Bruce, et al. (1992). Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. Proceedings of the SIGCHI conference on Human factors in computing systems. Monterey, California, United States, ACM Press.

Froehlich, J. and P. Dourish (2004). Unifying Artifacts and Activities in a Visual Tool for Distributed Software Development Teams. Proceedings of the 26th International Conference on Software Engineering, IEEE Computer Society.

Ganoe, C. H., J. P. Somervell, et al. (2003). Classroom BRIDGE: using collaborative public and desktop timelines to support activity awareness. Proceedings of the 16th annual ACM symposium on User interface software and technology. Vancouver, Canada, ACM Press.

Goldberg, J. and K. Mochel (2006). Developing Enterprise Application Design Guidelines for Large and Multiple Displays CHI. Montreal, Canada.

Greenberg, S. and M. Rounding (2001). The notification collage: posting information to public and personal displays. Proceedings of the SIGCHI conference on Human factors in computing systems. Seattle, Washington, United States, ACM Press.

Griswold, W. G., J. J. Yuan, et al. (2001). Exploiting the map metaphor in a tool for software evolution. Proceedings of the 23rd International Conference on Software Engineering. Toronto, Ontario, Canada, IEEE Computer Society.

Grudin, J. (2001). Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. Proceedings of the SIGCHI conference on Human factors in computing systems. Seattle, Washington, United States, ACM Press.

Henderson, J. and S. Card (1986). "Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface." ACM Trans. Graph. **5**(3): 211-243.

Hutchings, D. R. and J. Stasko (2007). Consistency, multiple monitors, and multiple windows. Proceedings of the SIGCHI conference on Human factors in computing systems. San Jose, California, USA, ACM.

Jiang, H., E. Ofek, et al. (2006). Direct pointer: direct manipulation for large-display interaction using handheld cameras. Proceedings of the SIGCHI conference on Human Factors in computing systems. Montr\&\#233;al, Qu\&\#233;bec, Canada, ACM.

Jones, J. A., M. J. Harrold, et al. (2002). Visualization of test information to assist fault localization. Proceedings of the 24th International Conference on Software Engineering. Orlando, Florida, ACM Press.

Karam, M. (2006). PhD Thesis: A framework for research and design of gesture-based human-computer interactions. ECS, University of Southampton. **PhD**.

Kruchten, P. (2003). Rational Unified Process An Introduction, Addison Wesley Professional.

Lommerse, G., F. Nossin, et al. (2005). The Visual Code Navigator: An Interactive Toolset for Source Code Investigation. Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization, IEEE Computer Society.

Marcus, A., L. Feng, et al. (2003). 3D representations for software visualization. Proceedings of the 2003 ACM symposium on Software visualization. San Diego, California, ACM Press.

McCrickard, D. S., C. M. Chewar, et al. (2003). "A model for notification systems evaluation—assessing user goals for multitasking activity." ACM Trans. Comput.-Hum. Interact. **10**(4): 312-338.

Newman, W. M. (1997). Better or just different? On the benefits of designing interactive systems in terms of critical parameters. Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques. Amsterdam, The Netherlands, ACM Press.

Ni, T., D. A. Bowman, et al. (2006). Increased display size and resolution improve task performance in Information-Rich Virtual Environments. Proceedings of the 2006 conference on Graphics interface. Quebec, Canada, Canadian Information Processing Society.

Nienaber, R. and E. Cloete (2003). A software agent framework for the support of software project management. Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology, South African Institute for Computer Scientists and Information Technologists.

North, C. and B. Shneiderman (2000). "Snap-together visualization: can users construct and operate coordinated visualizations?" International Journal of Human-Computer Studies **53**(5): 715-739.

Orso, A., J. A. Jones, et al. (2004). Gammatella: Visualization of Program-Execution Data for Deployed Software. Proceedings of the 26th International Conference on Software Engineering, IEEE Computer Society.

Padayachee, K. (2002). An interpretive study of software risk management perspectives. Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology. Port Elizabeth, South Africa, South African Institute for Computer Scientists and Information Technologists.

Renieris, M. and S. P. Reiss (1999). Almost: exploring program traces. Proceedings of the 1999 workshop on new paradigms in information visualization and manipulation in conjunction with the eighth ACM internation conference on Information and knowledge management. Kansas City, Missouri, United States, ACM Press.

Ringel, M. (2003). When one isn't enough: an analysis of virtual desktop usage strategies and their implications for design. CHI '03 extended abstracts on Human factors in computing systems. Ft. Lauderdale, Florida, USA, ACM.

Su, R. E. and B. P. Bailey (2005). Put Them Where? Towards Guidelines for Positioning Large Displays in Interactive Workspaces. INTERACT 2005, Rome, Italy.

Swaminathan, K. and S. Sato (1997). "Interaction design for large displays." interactions **4**(1): 15-24.

Tan, D. S. (2004). Exploiting the cognitive and social benefits of physically large displays, Carnegie Mellon University**:** 187.

Tan, D. S. and M. Czerwinski (2003). Information voyeurism: social impact of physically large displays on information privacy. CHI '03 extended abstracts on Human factors in computing systems. Ft. Lauderdale, Florida, USA, ACM.

Tan, D. S., D. Gergle, et al. (2003). With similar visual angles, larger displays improve spatial performance. Proceedings of the SIGCHI conference on Human factors in computing systems. Ft. Lauderdale, Florida, USA, ACM Press.

Tani, M., M. Horita, et al. (1994). Courtyard: integrating shared overview on a large screen and per-user detail on individual screens. Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence. Boston, Massachusetts, United States, ACM Press.

Tessendorf, D., C. M. Chewar, et al. (2002). An ordering of secondary task display attributes. CHI '02 extended abstracts on Human factors in computing systems. Minneapolis, Minnesota, USA, ACM Press.

Voinea, L., A. Telea, et al. (2004). EZEL: a Visual Tool for Performance Assessment of Peer-to-Peer File-Sharing Network.

Voinea, L., A. Telea, et al. (2005). CVSscan: visualization of code evolution. Proceedings of the 2005 ACM symposium on Software visualization. St. Louis, Missouri, ACM Press.

Wijk, J. J. v. and H. v. d. Wetering (1999). Cushion Treemaps: Visualization of Hierarchical Information. In proceedings of INFOVIS'99, San Francisco.

Wongthongtham, P., E. Chang, et al. (2004). Ontology-based multi-agent system to multi-site software development. Proceedings of the 2004 workshop on Quantitative techniques for software agile process. Newport Beach, California, ACM Press.