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
Chris North, Member
Ayman Abdel-Hamid, Member

June 5, 2007
Blacksburg, Virginia

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

© Copyright 2007, Khaled Hussein
Towards the Development of User Interface Design Guidelines for Large Shared Displays

Khaled Hussein

ABSTRACT

As large displays become more affordable, researchers investigate their effects on productivity and try to develop techniques for making the large display user experience more effective. Recent work has demonstrated significant productivity benefits, but has also identified numerous usability issues with current software design not scaling well. Studies show that large displays enable users to create and manage more windows, as well as to engage in more complex multitasking behavior. In this thesis, we developed some user interface design guidelines for large shared displays.

Specifically, empirical studies to compare the effects of using large shared displays against personal displays when each of them is used as a secondary display are presented, showing that large shared display impose higher interruption and comprehension to the user. 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 LSD along with the guidelines to further enhance and improve team efficiency and productivity in collaborative software development environments. Finally, an in-situ evaluation assesses the benefits of SuperTrack based on our developed design guidelines in terms of improving software development efficiency and productivity. Results show that by exposing software development team members to a large shared display, a system that follows our developed user interface guidelines leads to higher communication among the team members and improved group awareness, leading to higher 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, you supported me through all my bad days, and you 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, advisor, for his incredible help and guidance as well as for his constant availability and dedication to quality. Thank you for making this entire research effort such a challenging, shaping, and influential yet enjoyable experience.

It is a great pleasure to acknowledge here 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

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>xi</td>
</tr>
<tr>
<td><strong>Chapter 1</strong> INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Problem Description</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Motivation</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Thesis Statement</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Research Approach</td>
<td>4</td>
</tr>
<tr>
<td>1.5 Contributions and Anticipated Impact</td>
<td>5</td>
</tr>
<tr>
<td>1.6 Overview of the Thesis</td>
<td>7</td>
</tr>
<tr>
<td><strong>Chapter 2</strong> RELATED WORK</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Research in Dual-Task Situations</td>
<td>9</td>
</tr>
<tr>
<td>2.1.1 Dual-Task Concept</td>
<td>10</td>
</tr>
<tr>
<td>2.1.2 Attention utility theme research</td>
<td>12</td>
</tr>
<tr>
<td>2.1.3 Framework for Understanding Notification Systems</td>
<td>15</td>
</tr>
<tr>
<td>2.2 Personal and Shared Displays</td>
<td>19</td>
</tr>
<tr>
<td>2.2.1 Types of personal displays</td>
<td>19</td>
</tr>
<tr>
<td>2.2.2 Large Displays in Real Life</td>
<td>21</td>
</tr>
<tr>
<td>2.3 Using Large Shared Displays in Interactive Workspaces</td>
<td>28</td>
</tr>
<tr>
<td>2.3.1 Guidelines for using small/personal and large/shared displays</td>
<td>28</td>
</tr>
<tr>
<td><strong>Chapter 3</strong> SECONDARY TASKS: HOW CLOSE SHOULD THEY BE?</td>
<td>32</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>33</td>
</tr>
<tr>
<td>3.2 Experiment 1</td>
<td>34</td>
</tr>
<tr>
<td>3.2.1 Hypotheses</td>
<td>34</td>
</tr>
<tr>
<td>3.2.2 Experimental Design</td>
<td>34</td>
</tr>
<tr>
<td>3.2.3 Participants</td>
<td>36</td>
</tr>
<tr>
<td>3.2.4 Materials</td>
<td>37</td>
</tr>
<tr>
<td>3.2.5 Tasks</td>
<td>37</td>
</tr>
<tr>
<td>3.2.6 Procedure</td>
<td>38</td>
</tr>
<tr>
<td>3.2.7 Results</td>
<td>39</td>
</tr>
<tr>
<td>3.2.8 Summary</td>
<td>41</td>
</tr>
<tr>
<td>3.3 Experiment 2</td>
<td>42</td>
</tr>
<tr>
<td>3.3.1 Hypotheses</td>
<td>42</td>
</tr>
<tr>
<td>3.3.2 Experimental Design</td>
<td>42</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>3.3.3 Participants</td>
<td>43</td>
</tr>
<tr>
<td>3.3.4 Tasks</td>
<td>43</td>
</tr>
<tr>
<td>3.3.5 Procedure</td>
<td>45</td>
</tr>
<tr>
<td>3.3.6 Results</td>
<td>45</td>
</tr>
<tr>
<td>3.3.7 Summary</td>
<td>46</td>
</tr>
<tr>
<td>3.4 Conclusion</td>
<td>46</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>INFORMATION DESIGN GUIDELINES</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>49</td>
</tr>
<tr>
<td>4.2 Source Code Visualizations</td>
<td>51</td>
</tr>
<tr>
<td>4.3 SYSTEM 1: VISUAL CODE NAVIGATOR (VCN)</td>
<td>52</td>
</tr>
<tr>
<td>4.3.1 Implementation</td>
<td>52</td>
</tr>
<tr>
<td>4.3.2 Views and Interaction</td>
<td>52</td>
</tr>
<tr>
<td>4.4 SYSTEM 2: Augur</td>
<td>55</td>
</tr>
<tr>
<td>4.4.1 Implementation</td>
<td>55</td>
</tr>
<tr>
<td>4.4.2 Views and Interaction</td>
<td>56</td>
</tr>
<tr>
<td>4.5 Experiment 3</td>
<td>58</td>
</tr>
<tr>
<td>4.5.1 Hypotheses</td>
<td>58</td>
</tr>
<tr>
<td>4.5.2 Experimental Design</td>
<td>59</td>
</tr>
<tr>
<td>4.5.3 Participants</td>
<td>60</td>
</tr>
<tr>
<td>4.5.4 Procedures</td>
<td>60</td>
</tr>
<tr>
<td>4.5.5 Results</td>
<td>60</td>
</tr>
<tr>
<td>4.5.6 Summary</td>
<td>61</td>
</tr>
<tr>
<td>4.6 Interviews</td>
<td>61</td>
</tr>
<tr>
<td>4.6.1 Results</td>
<td>62</td>
</tr>
<tr>
<td>4.7 Guidelines</td>
<td>64</td>
</tr>
<tr>
<td>4.7.1 Guideline 1</td>
<td>65</td>
</tr>
<tr>
<td>4.7.2 Guideline 2</td>
<td>66</td>
</tr>
<tr>
<td>4.8 Conclusion</td>
<td>66</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>SUPERTRACK: IMPLEMENTATION DETAILS AND USER STUDY</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>68</td>
</tr>
<tr>
<td>5.2 Exploring Group Awareness Needs</td>
<td>71</td>
</tr>
<tr>
<td>5.3 Designing to Enhance Group Awareness</td>
<td>74</td>
</tr>
<tr>
<td>5.3.1 Time Tracking View</td>
<td>74</td>
</tr>
<tr>
<td>5.3.2 Project Tracking View</td>
<td>77</td>
</tr>
<tr>
<td>5.3.3 Dependencies View</td>
<td>79</td>
</tr>
<tr>
<td>5.4 In-Situ Study</td>
<td>81</td>
</tr>
<tr>
<td>5.4.1 Hypotheses</td>
<td>81</td>
</tr>
<tr>
<td>5.4.2 Study Overview</td>
<td>82</td>
</tr>
<tr>
<td>5.4.3 Participants</td>
<td>82</td>
</tr>
<tr>
<td>5.4.4 Materials</td>
<td>83</td>
</tr>
<tr>
<td>5.4.5 Procedure</td>
<td>84</td>
</tr>
<tr>
<td>5.4.6 Results and Discussion</td>
<td>84</td>
</tr>
<tr>
<td>5.4.7 Summary</td>
<td>89</td>
</tr>
</tbody>
</table>
5.5 Conclusion

Chapter 6  CONCLUSION AND FUTURE WORK  91

6.1 Summary of the Work  91
6.2 Major Findings and Contributions  92
6.3 Future Work  94

Chapter 7  APPENDICES  96

7.1 Appendix A  97
  7.1.1 First Round  97
  7.1.2 Second Round  99
7.2 Appendix B  102
  7.2.1 First experiment results  102
  7.2.2 Second experiment results  104

REFERENCES  106
LIST OF FIGURES

Figure 1: The IRC notification-system framework modeling the notification-system classes according to the notification goals in terms of interruption, reaction, and comprehension ................................................................. 16

Figure 2: The NCSA Display Wall-In-A-Box is a 20 projector array carefully stitched together to form a single large high-resolution display ........................................................................................................................................ 22

Figure 3: Kimura presents interactive montages of images on large peripheral displays to help users manage multiple working tasks ........................................................................................................................................ 23

Figure 4: The SMART Board provides a large touch screen display that supports (left) informal group meetings or (right) presentations ........................................................................................................................................ 26

Figure 5: The setup of the primary and secondary displays. The personal display (close) and the large display (far) are in the same field of view ........................................................................................................................................ 36

Figure 6: The large shared display is a projected display in the same field of view as the personal display ........................................................................................................................................ 37

Figure 7: Main effect of secondary display type. Users performed significantly better when personal display was used rather than large shared display ........................................................................................................................................ 39

Figure 8: 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 ........................................................................................................................................ 44

Figure 9: VCN Syntactic View ........................................................................................................................................................................................................................................................................................................................................ 53

Figure 10: VCN Symbol View ........................................................................................................................................................................................................................................................................................................................................ 54

Figure 11: VCN Evolution View ........................................................................................................................................................................................................................................................................................................................................ 55

Figure 12: Augur Overview ........................................................................................................................................................................................................................................................................................................................................ 56
Figure 13: Augur History View ...........................................................................................................57

Figure 14: Augur Author View ...........................................................................................................58

Figure 15: Hierarchical Task Analysis for the software development process followed in a big software development company ........................................................................................................72

Figure 16: Time tracking view showing that member M2 is not assigned activity A1 .......................76

Figure 17: Time tracking view showing that member M3 finished all his work and is considered an empty resource that should be utilized ................................................................................77

Figure 18: Project tracking view showing the progress of each activity along with the members assigned to work in each activity ................................................................................................78

Figure 19: Dependencies view showing the dependencies of the project, discriminating internal dependencies with black color, and external dependencies with red color ................................................80

Figure 20: SuperTrack overview with its three views ........................................................................81

Figure 21: The room arrangement of the experiment ........................................................................84
LIST OF TABLES

Table 1: Attention benefits. Notification system users expect to gain benefits associated with fulfillment of user goals by sacrificing attention from other tasks. ........................................... 13

Table 2: Attention costs. Costs can be exacerbated by factors of the current situation ............... 14

Table 3: Participants roles in the in-Situ study .................................................................................. 83
This chapter describes the problem of interest as studying the differences between personal and large shared displays, and developing interface design guidelines for large shared displays. The motivation for this research comes from the increased usage of large shared displays in various work environments and the emerging need for enhanced performance in collaborative environments. Our contributions are comparing personal and large secondary displays with respect to critical parameters of notification systems, developing interface guidelines for large secondary displays, and implementing a system that recruits our design guidelines with the intention of enhancing group awareness in a collaborative software development environment.

1.1 Problem Description

Most of our daily activities are divided between the real world and the digital world. While the real world is represented by the physical environment surrounding us, the digital world is represented by the digital reality that we interact with. The two worlds are believed to form an integral part of how we think and act (Hollan, Hutchins, and Kirsh, 2000). Therefore, as we explore techniques that support human-computer interaction, we
should expand our focus to include these larger environments: how they augment the way we think or work, how they affect the way we interact with other people around us, and how we can best design these worlds to create more productive work environments.

In multiple-task scenarios, the user is engaged in a primary task while attending to one or more secondary tasks. At any given time, most of the information in our environment is peripheral to our main focus of attention, which leads to multiple-task situations. In fact, the need for many forms of peripheral information increases as other primary tasks occur. For example, the desire of users to appraise road and traffic conditions increases as soon as they begin driving. This example illustrates the common challenge in designing secondary tasks, which explores how a user can obtain the desired utility from a secondary task while not compromising the goals for the primary one. Empirical research revealed that even slightly modifying the way information is presented can affect user attention considerably, leading to somewhat different notification goals (McCrickard and Chewar, 2003). Since user attention is a limited resource, designers need to ascertain whether and how much interruption users should receive and whether users have to stop their current task immediately or at a later point to react to a notification. Therefore, the delivery of content in multiple-task situations has to be carefully considered on a peripheral level, presenting a greater challenge to designers.

With the increased diversity of displays comes the challenge of how and when to use different displays appropriately to support secondary tasks. For example, large shared displays would be appropriate in situations where overall system status may help an individual make a better-informed decision regarding his/her specific task. In cases where summary information is not needed for an operator to complete his/her tasks, the large shared displays could actually detract from the individual’s performance. Therefore, our first question of interest is to explore the effects of large shared displays on the different tasks that the user involved in.

If it is determined that large shared displays are appropriate for the situations at hand, there are multiple issues that designers should consider prior to implementation, most importantly including the display’s size and configuration, the information presented on
the large shared displays, consistency with the rest of the design, and control of the views. Therefore, we see an urgent need for developing guidelines for the use of large shared displays in collaborative environments. In addition, designers need interface design guidelines to convey information on large shared displays, leading to maximized individual/group performance.

1.2 Motivation

We believe that computer displays serve as bridges that connect the real and virtual worlds. Computer displays possess a certain duality since they exist in the real world while allowing us to peer into the digital one. In a study of multiple monitor usage in the workplace (Grudin, 2001), Grudin found that most users divide information between the monitors in such a way that the focal monitor is used for main tasks while the peripheral monitor is used for secondary tasks. Thus, understanding the role of physically large displays is significant because of the emerging trend in the workplace towards multiple display systems that have the potential to provide abundant display space distributed throughout the environment. Such workplaces typically include several types of displays, each with characteristics that may make it more or less suitable for certain tasks. Within these workplaces, large displays are becoming prevalent.

Large shared displays are ubiquitous in work and social places and the use of such display surfaces to support collaborative work has been a major theme of research and practice in the Computer Supported Collaborative Workspaces (CSCW) and HCI communities. We note that while collaborative work takes many forms and occurs in wide ranging settings, many of these technological developments are concerned with very specific forms of collaboration such as supporting awareness, remote collaboration, design and scheduling tasks and knowledge work. Therefore, researchers explore the use of large shared displays to provide overall, group-view information. Although it is necessary to consider multiple factors when determining the added value of introducing large shared displays, the application should be given primary consideration. Depending on the final application, large shared displays can be implemented to provide some of the following capabilities: (1) display current system status to multiple team members
simultaneously, (2) provide a “walk-up” display for briefing or group-work purposes, (3) enable team members to see the effects of their actions on the tasks of other operators, and (4) facilitate monitoring of the team by the leader (Roth et al., 1998). In conclusion, it has been found that large displays enable juxtaposition of alternative ideas (Biehl and Bailey, 2004), support awareness of peripheral information (Tan and Czerwinski, 2003), and enable more effective discussion of digital information (Streitz et al., 1999).

Generally, peripheral displays need to address two sets of attention issues (Matthews et al., 2004). One deals with context about the user, including interruptibility (Fogarty, Hudson, and Lai, 2004; Horvitz and Apacible, 2003), current primary activity or task (Dey, Salber, and Abowd, 2001; Horvitz and Apacible, 2003), and focus of attention (Horvitz and Apacible, 2003). The other one deals with attention management, or how a display can convey information appropriately. To utilize user’s attention while presenting information on peripheral displays, designers need a set of guidelines to follow while designing peripheral information on large shared displays.

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 and large shared ones even at identical visual angle. Understanding these differences leads to developing a set of guidelines which draws designers’ attention to 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

We believe that a balance is needed between staying on the path towards a larger vision, and exploring the sometimes tangential but often interesting problems that inevitably arise along that path. The former ensures that researchers do not get lost wandering the
design space looking for inconsequential problems to solve. The latter ensures that we do not become totally engrossed in a single problem, which might lead us to lose sight of the greater goals and vision that drive our higher level agenda.

Much of this work has been motivated by a combination of anecdotal evidence, informal observations, and established theoretical work in psychology, human-computer interaction, and computer science. It is through these channels that we were able to identify and focus on the areas in which we thought large displays were likely to have the most impact on user performance. The general approach was a three-pronged strategy including: (1) combining theoretical work with empirical evidence to identify display characteristics most likely to impact the way we work, (2) designing controlled experiments to isolate and understand effects more completely, and (3) deriving design guidelines and building real-world systems that make users more productive.

1.5 Contributions and Anticipated Impact

To achieve maximum productivity, there must be an ideal level of interruption from the primary task, reaction to the notification sent, and comprehension of information presented by the notification or the secondary task. A conceptual model recognizes three critical parameters: Interruption, Reaction, and Comprehension, to model user’s notification goals (Chewar, McCrickard, and Sutcliffe, 2004). Interruption is defined as an event triggering a reallocation of attention from a task to the notification. Reaction is characterized as the rapid and accurate response to the stimuli the notification system produces. Finally, comprehension is referred to as the objective of making sense and recollecting the conveyed information at a later time (McCrickard, Chewar, Somervell, and Ndiwalana, 2003). These three rationales serve as aces for a framework to represent the degree of benefits on users’ notification goals resulting from their use.

The critical-parameter levels of desirable user Interruption, Reaction, and Comprehension (IRC) are expressed as an ordinal set of numerical values ranging from zero to one. A value of zero depicts a low objective level, while a value of one corresponds to a high notification assessment for the considered notification parameter. The establishment of
this conceptual model allows designers to capture accurately and numerically the notification goal of the user (McCrickard and Chewar, 2003). Based on this framework, we initiated our research by studying the impacts of using different types of secondary displays. The contributions of our work and the anticipated impacts in real life and future research are presented next.

“Comparison between personal and public peripheral displays in terms of interruption and comprehension”

In this work, we present empirical studies demonstrating the differences between secondary personal display and secondary public counterpart in terms of interruption and comprehension. The studies document a higher interruption received from the use of secondary large displays over personal ones. Similarly, a higher comprehension is noticed, which shows promising opportunities benefited by shifting peripheral information to secondary large displays.

“Interface design guidelines for shared peripheral displays”

Our investigation to the effect of large shared displays on user’s performance in primary and secondary tasks shows that there are significant performance 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 improving group awareness in such collaborative software development environment is implemented and presented on the 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. Kirsh classifies the functions of physical space into three categories: spatial arrangements that simplify choice, spatial arrangements that simplify perception, and spatial dynamics that simplify internal computation (Kirsh, 1995). This work simplifies and clarifies the tradeoffs designers should make in order to design the physical space of a collaborative work environment. Specifically, this work also provides interface designers with a set of guidelines to follow when designing or shifting information into large secondary displays. Finally, SuperTrack demonstrates a system for enhancing group awareness in collaborative software development environments.

1.6 Overview of the Thesis

In this chapter, we briefly presented a high level description of the problem at hand, the motivation, research approach followed, and contributions. In the following, a description of the thesis organization is presented.

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 dual-task situations and how researchers tend to meet the attention management challenges. Moreover, it presents the previous research contributed to designing user interfaces for peripheral task on large shared displays. The work presented within this chapter forms the foundation for the work in this thesis.

Chapter 3 presents a series of experiments showing that large shared displays, even when viewed at identical visual angles to small or personal ones, 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 in light of critical parameters such as Interruption and Comprehension.

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

In chapter 5, SuperTrack, a software projects tracking tool, is presented along with its motivation, implementation details, and experimental results of our in-situ study. Moreover, the chapter applies the previous work of developed interface design guidelines to the real world.

Finally, in chapter 6, the work and contributions presented in this work are summarized. In addition, directions for future work are also explored in this chapter.

Key materials from the main experiments in this thesis are included within an appendix that follows.
This chapter provides a literature review of the research related to the work presented in this thesis in three parts. The first part begins by describing the meaning of dual-task situations, the relationships between primary and secondary tasks, the attention-utility theme concerning designing dual-task scenario, and a discussion of evaluating notification systems as an example of secondary tasks. The second part covers the types of personal and shared displays along with the most important applications of large shared displays in different work environments. The third part discusses the previous research that focused on how to use large displays in interactive workspaces. Finally, the chapter provides coverage of the most crucial guidelines for using and designing interfaces for large shared displays, showing the need to develop more specific guidelines related to the use of large shared displays in collaborative environments.

2.1 Research in Dual-Task Situations

While graphical design principles form a large body of literature for information design objects in a user’s focal attention, the interfaces within our concern (notification systems) are usually used outside focal attention as peripheral information. Research on these
Research in Dual-Task Situations

systems is emerging as an area of interest within HCI, but work within the human factors and experimental psychology fields have addressed similar issues for years. Perhaps the most comprehensive review can be found in Wickens and Hollands’ discussion of dual-task situations, in which they review primary task performance degradation in terms of resource allocation to secondary tasks and adaptation consequences for excessive workloads (Wickens and Hollands, 2000). Here we summarize the most important ideas relating to dual-task situations.

2.1.1 Dual-Task Concept

In a dual-task situation, there is typically one task that receives attention emphasis, which is referred to as the primary task, while the concurrent task is referred to as the secondary task. This is not to say that goals corresponding to the primary task are more important or urgent than those of the secondary task. Wikens discusses two methods of attention division between tasks: graded and discrete (Wickens and Hollands, 2000). In graded resource allocation, a portion of attention is consistently devoted to each task, with the primary task receiving a higher portion. Discrete allocation splits a given period of time (presumably on the order of seconds or minutes) into blocks in which each task receives focused attention; of course, the primary task would receive attention for more time within the given period. Nominally, we assume that notification systems correspond to secondary tasks for which attention is allocated on a discrete basis.

Basically, when a user’s attention is divided between tasks, forcing concurrent processing or time sharing, this is a dual-task situation (we presume one of these tasks to involve the use of a notification system). If both tasks can be performed simultaneously as well as they could be performed independently, then these tasks are not relevant and the notification system can be evaluated as any other focal display or interface. However, we believe that the use of all but the simplest systems will cause some performance degradation in the other tasks, implying applicability of the attention-utility theme, which is explained later, and of the evaluation strategies presented in this work.
Other than fundamental differences in possible primary-secondary task relationships, which are discussed next, many other factors may impact dual-task performance and our ability to understand successful aspects of supporting information design. Some of these include total mental workload required by the system, presence of data linked task dependencies, and differences among users (Wickens and Hollands, 2000). There are certainly conceivable cases of dual-task situations, particularly when one of both tasks is complex or urgent, which would result in overload, meaning that mental resources are fully consumed and expected performance levels cannot be satisfied and/or diminish over time. However, other mental workload characteristics may lead to boredom and disinterest, or attention resources may be completely consumed by the dual-task goals. In dual-task situations, task execution could be constrained at certain points by availability of data, implying that only so much processing or interaction can be performed (representing progress toward a goal) within a given period of time. Lastly, users certainly have differences in skill levels, perceptual capabilities, and context-switching ability, differences which are important in all areas of HCI but may have even larger implications for understanding effectiveness of notification systems.

**Primary-Secondary Task Relationships**

When the relationships between the primary and secondary tasks are considered, there are a few important factors which may impact design and evaluation of notification system, including the relationship between the two task goals and the nature of the tasks. The two tasks can have goals that are unrelated or that are dependent on information presented in the other. For example, users may want to remain notified of weather conditions while they edit a document, two tasks with unrelated goals. However, in a different usage scenario, information in an instant messaging or collaborative status reporting system could prompt a user to transition from a document creation primary task to a spreadsheet modification task as primary. In other cases, secondary information may not cause a shift in primary task goals, but it may influence primary task execution decisions. Ubiquitous navigation systems are a good example of this type of secondary task: based on the route information presented a vehicle operator may perform a driving task differently.
Each task may also have intrinsic properties that affect the other task and also affect expectations of notification system performance. High levels of user proficiency or repetitive actions may allow one or both tasks to become automatic, requiring less attention devoted to achieve a desired level of performance. Wickens and Hollands describe other task structural factors, such as modality and resource consumption, which may place unusual strain on a user’s attention capacity. Specifically, time-sharing efficiency is enhanced when task modalities allow parallel perception (referred to as cross-modality) rather than less efficient, intra-modal perception that could result in serial processing or signal confusion (e.g., reading a book while driving is harder than listening to the same book on the vehicle’s stereo system while driving) (Wickens and Hollands, 2000). A final concept from Wickens’ discussion considers simultaneous consumption of processing resources (perception, working memory, and response) by each task. When both tasks require use of the same human cognition resources, aspect of dual-task performance may decline although no shift in attention occurred.

2.1.2 Attention utility theme research

Notification systems are defined as interfaces that are typically used in a divided-attention, multitask situation, attempting to deliver currently valued information through a variety of platforms and modes in an efficient and effective manner (McCrickard and Chewar, 2003). McCrickard and Chewar also presented an important distinction between notification systems and traditional HCI research, which they call the attention-utility theme (McCrickard and Chewar, 2003), asserting that it is useful to think as a constrained resource that can be traded for some utility. This utility is enabled by perceiving additional valued information while performing other tasks. This attention-utility tradeoff can be stated as “The success of a notification system hinges on accurately supporting attention allocation between tasks, while simultaneously enabling utility through access to additional information” (Chewar, 2005).

The attention-utility theme concisely captures the source of scarcity (the attention of the use) along with the user’s purpose in using the notification system (utility associated with access to an additional source of information). Certainly this relationship is not smooth
and differentiable, but still generally describes the cost of achieving user goals—a cost which reliably yields benefits when the state of a user’s attention can be modeled and matched with appropriate information rendering.

<table>
<thead>
<tr>
<th>Usability Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Goals</strong></td>
</tr>
<tr>
<td>• Pace daily activities</td>
</tr>
<tr>
<td>• Prompt task transition</td>
</tr>
<tr>
<td>• Receive urgent/timely information</td>
</tr>
<tr>
<td>• Synchronize with colleagues</td>
</tr>
<tr>
<td>• Make decisions</td>
</tr>
<tr>
<td>• Modify primary task approach</td>
</tr>
<tr>
<td>• Provide response</td>
</tr>
<tr>
<td>• Acknowledge status</td>
</tr>
<tr>
<td>• Identify state changes</td>
</tr>
<tr>
<td>• Understand patterns and trends</td>
</tr>
<tr>
<td>• Assimilate complex information monitor</td>
</tr>
<tr>
<td>• Resources over time gain awareness</td>
</tr>
<tr>
<td>• Reduce stress</td>
</tr>
<tr>
<td>• Emote humor</td>
</tr>
<tr>
<td>• Cultivate enjoyment</td>
</tr>
<tr>
<td>• Augment meaning or presence</td>
</tr>
<tr>
<td>• Increase feeling of security</td>
</tr>
</tbody>
</table>

Table 1: Attention benefits. Notification system users expect to gain benefits associated with fulfillment of user goals by sacrificing attention from other tasks.

Table 1 itemizes component cost-benefit factors of the attention-utility tradeoff. Users ultimately use a notification system to gain benefits, which come from specific types of utility. Four general sources of utility are recognized. These sources can result from associated user goals. The general goals of comprehension, reaction, and interruption can be thought of as critical parameters-key measures of system success that can be benchmarked to reveal design progress. These goals are unique in that the user is willing to sacrifice a certain amount of primary task attention in order to achieve them. Other important system features and user needs must be typically supported in user interfaces in order to include privacy, reliability, and trust. These features can negatively influence the
amount of required attention without providing a distinct benefit that independently motivates system use.

### Table 2: Attention costs

Costs can be exacerbated by factors of the current situation.

<table>
<thead>
<tr>
<th>Situation Parameter</th>
<th>Cost Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Context</strong></td>
<td>• Goal relationships of tasks</td>
</tr>
<tr>
<td></td>
<td>• Task perceptual-motor qualities</td>
</tr>
<tr>
<td></td>
<td>• Data-link dependencies</td>
</tr>
<tr>
<td></td>
<td>• Relative tasks priorities</td>
</tr>
<tr>
<td></td>
<td>• Interruptability</td>
</tr>
<tr>
<td></td>
<td>• Focus/peripheral location</td>
</tr>
<tr>
<td></td>
<td>• Platforms and environment</td>
</tr>
<tr>
<td><strong>User Characteristics</strong></td>
<td>• Skill and automaticity</td>
</tr>
<tr>
<td></td>
<td>• Cognitive and perceptual abilities</td>
</tr>
<tr>
<td></td>
<td>• Current overall mental workload</td>
</tr>
<tr>
<td></td>
<td>• Sender/receiver roles</td>
</tr>
<tr>
<td></td>
<td>• Demographics</td>
</tr>
<tr>
<td><strong>Information Characteristics</strong></td>
<td>• Granularity</td>
</tr>
<tr>
<td></td>
<td>• Discrete/continuous</td>
</tr>
<tr>
<td></td>
<td>• Modality (visual or auditory)</td>
</tr>
<tr>
<td></td>
<td>• Complexity</td>
</tr>
<tr>
<td></td>
<td>• Representation richness</td>
</tr>
<tr>
<td></td>
<td>• Anticipated value</td>
</tr>
<tr>
<td></td>
<td>• Synchronization</td>
</tr>
<tr>
<td></td>
<td>• Content relevance</td>
</tr>
</tbody>
</table>

The level of cost, determined by the amount of attention removed from ongoing tasks, may be elevated as a result of the factors presented in Table 2. For example, above average attention cost factors may include a user’s lack of skills in perceiving unfamiliar or complex notification information. Unfortunately, cost factors may not carry a constant value across different situations or result in expected benefits. Poor designs may result in a user accepting a certain cost in anticipation of a certain utility without actually receiving that utility. Usually, the attention required for a user to perceive and process a notification is diverted from attention focus on a primary task, but cost only results if primary task performance is negatively impacted. Attention supplied during natural breaks in a primary task can minimize cost. The many cost considerations and strategies
to reduce them amplify the importance of inferring and leveraging the state of a user’s attention and the semantic value of the notification for interface design.

2.1.3 Framework for Understanding Notification Systems

Discussing notification systems in a cohesive framework, defined by critical parameters, allows for effective evaluation and comparison. This fact stems from the fact that critical parameters capture the overarching goals of a system class, not just those for a single system. Instead of focusing evaluation on metrics derived from developer expectations, critical parameters provide grounded, reusable, and comparable metrics where evaluation is focused on determining if new systems provide advancements. In fact, critical parameters provide the criteria for establishing long term performance measures so that we can assess whether new systems are “better” or “just different” (Newman, 1997). The following presents a framework for describing notification systems based on the notion of critical parameters. The thrust of this work supports our goal of evaluating large shared displays as notification tools.

**Critical Parameters**

William Newman put forth the idea of critical parameters for guiding design and strengthening evaluation in (Newman, 1997) as a solution to the growing disparity between interactive system design and separate evaluation. For example, consider airport terminals, where the critical parameter would be flight capacity per hour per day (Newman, 1997). All airport terminals can be assessed in terms of this capacity, and improving that capacity would invariably mean we have a better airport. Newman argues that by establishing parameters for application classes, researchers can begin establishing evaluation criteria, thereby providing continuity in evaluation that allows us “to tell whether progress is being made” (Newman, 1997).

**Evaluating notification systems using IRC framework**

Notification system theory attempts to identify user goals in terms of three different critical parameters, interruption, reaction, and comprehension of the secondary task.
Interruption denotes reallocation of attention from the primary task to the secondary one. The interruption goals inherent in a notification system can and should be used when designing notification systems for secondary tasks. If interruption is a user goal present with a secondary task, the attention shift provides an opportunity to support a more accommodating environment for interaction to take place (McCrickard, Chewar, Somervell, and Ndiwalana, 2003). Reaction denotes the accuracy and urgency with which a response to a particular notification is needed. A faster response time with a particular input method will allow the user goal of quick reaction to be met, but it would not make sense to use a slower input method for the opposite case where only gradual reaction is called for. There are, however, cases where slower reaction time can allow for other beneficial tradeoffs (McCrickard, Chewar, Somervell, and Ndiwalana, 2003). While rapid and accurate reaction to an informational cue is important in many situations, often it is also vital to use notification systems with the goal of remembering the information for at least a brief period of time. Therefore, comprehension denotes making sense and
reallocating the conveyed information at a later time. Figure 1 provides a graphical representation for the notification-system framework categorization. This design space recognizes eight notification-system classes (McCrickard, Chewar, Somervell, and Ndiwalana, 2003), corresponding to each triplet combination of high (1) and low (0) levels for each of the interruption, reaction, and comprehension parameter:

1. Noise: IRC (0, 0, 0) refers to a class of notification systems where low interruption, reaction, and comprehension levels are valued. Notification systems such as internet radio belong to that category. Users of such systems are not willing to reallocate much of their attention to the notification (low interruption). Similarly, users will typically not need to react to the notification (low reaction). Finally, minimal amount of information is conveyed (low comprehension).

2. Ambient Media: IRC (0, 0, 1) is a type of system that focuses on providing users with valued information through effortless awareness.

3. Indicator: IRC (0, 1, 0) denotes a category of systems that center around helping users to perform actions while maintaining low reallocation of attention with respect to the primary task. Long-term knowledge gain is not valued. In-vehicle GPS navigation systems and dashboard gauges are examples of this class of systems. While driving a car, people react to a rev counter indicating a high rpm by shifting up (high reaction). Drivers want to remain focused on the road as much as possible (low interruption) and do not value gaining knowledge about rpm trends (low comprehension).

4. Secondary display: IRC (0, 1, 1) corresponds to a class of systems that contributes to a long term knowledge gain without introducing loss of attention and simultaneously triggers user’s reactions. For example, Microsoft Research’s Sideshow (Cadiz, Venolia, Jancke, and Gupta, 2002) resides on user’s primary display in the form of a sidebar and makes use of peripheral awareness (low interruption) to convey valued information about e-mail’s status, colleagues’ availability, and traffic conditions. Typical users can monitor traffic at their own initiative and develop a long-term knowledge gain such as establishing the
relationship between traffic patterns, weather conditions, and time (high comprehension). Based on the traffic jam information displayed on the glanceable interface, users can react to current traffic conditions by deciding to stay or leave the office (high reaction).

5. Diversion: IRC (1, 0, 0) corresponds to notification systems that display solely an interrupting behavior. Triggering users’ reaction or conveying long-term knowledge gain is not desired. Microsoft Office Assistant’s occasional animated behavior corresponds to the notification nature of such systems. The animation catches users’ attention (high interruption), while they typically will not need to react to the event (low reaction). The animated behavior conveys minimal levels of information (low comprehension).

6. Information Exhibit: IRC (1, 0, 1) is a genre of systems that presents a high interruption goal but does not seek to trigger users reaction. Comprehension gain resulting from the notification is highly valued. A case in terms of such systems is Photo News Board. The system uses a matrix-like interface to display news stories in the form of pictures arranged by theme on a large screen (Somervell, Chewar, McCrickard, and Ndiwalana, 2003). Recently retrieved news stories appear in the center of the display while older stories shift toward the edges. The resulting motion drags user’s attention toward the display (high interruption). The system also highlights pictures of the news stories relevant to the room occupants. Community members use that system to fulfill their high comprehension needs (high comprehension). Users typically will not need to react to the event (low reaction).

7. Alarm: IRC (1, 1, 0) is a category of systems that exhibits high interruption to the primary task and where user reaction is highly valued, while the comprehension level remains minimal. Enabling users to redirect their activity is the critical aspect of such systems. In case of an emergency, people depend on a fire alarm to abruptly and fully gain their attention (high interruption) so that they can react quickly and evacuate the premises in a timely manner (high reaction). By nature,
an alarm does not seek to convey any information related to the causes of the incident, its location, or its progress (low comprehension).

8. Critical activity monitor: IRC (1, 1, 1) is a type of system wherein high interruption, high reaction, and high comprehension levels are valued. For example, Microsoft Research’s Scope (van Dantzich, Robbins, Horvitz, and Czerwinski, 2002) makes use of a radar metaphor to enable users to monitor the nature, status, and content of various crucial daily work activities such as email and instant messages, tasks, and appointments. This glanceable interface notifies users of incoming emails or overdue tasks, using motion to catch user attention (high interruption) and visual coding through shapes and colors to describe the email or the task nature (high comprehension). Users will typically react to an important email or an overdue task (high reaction).

2.2 Personal and Shared Displays

2.2.1 Types of personal displays

Personal displays can be considered the ideal solution for some of privacy problems. Therefore, there is an extensive research on exploring several issues related to personal displays. This section introduces and highlights some of the diverse types of personal displays.

Personal displays

The word “personal” imposes that the emerging desktop, handheld, and wearable technologies are geared toward support for individual users, with support for single-user tasks rather than collaborative activities. Therefore, for personal technologies to also become a utility for collaborative tasks, their networking is prerequisite.

There are several types of common personal displays. Desktop displays can be classified as the most common personal displays. Since desktop displays are most used in homes,
business, and several other environments, HCI researchers and designers focused on developing interface design guidelines for desktop displays.

Supporting portability, researchers were able to design portable personal computers such as laptops and Tablet PCs. With the increased sizes and nature of the portable personal displays come several interaction techniques. For example, displays in Tablet PCs uses stylus for pointing and touch screens.

**Handheld displays**

Powerful handheld devices are rapidly paving their way as people’s personal trusted devices. This trend is visible in the increasing capabilities of smartphones and PDAs, enabling these devices to be used for an ever-increasing variety of tasks. If the experts are on the mark, very soon handheld computer technology will supplant the desktop computer as ubiquitous technology on campuses and in the workplace (Weiser, 1998).

Personal Digital Assistants (PDAs) have a typical display size of 160x160 pixels in a touch based screen. The touch-sensitive screens in handhelds offer greater flexibility for software design, but their interfaces are traditionally designed for pen-based interaction requiring two hands (Parhi, Karlson, and Bederson, 2006).

**Uncommon personal displays**

The Virtual Retinal Display (VRD) is a new display technology that scans modulated low energy laser light directly onto the viewer’s retina to create a perception of a virtual image. VRD was invented at the University of Washington in the Human Interface Technology Lab (HIT) in 1991. The aim was to produce a full color, wide field of view, high resolution, high brightness, low cost virtual display. The VRD projects a modulated beam of light (from an electronic source) directly onto the retina of the eye producing a rasterized image. The user has the illusion of seeing the source image as if he stands two feet away in front of a 14-inch monitor (Lin, Seibel, and Furness, 2003).

A Head Mounted Display or Helmet Mounted Display (HMD) has either one or two small displays with lenses and semi-transparent mirrors embedded in a helmet, eye-
glasses, or visor (Kiyokawa, Billinghurst, Campbell, and Woods, 2003). A HMD has the potential to display a different image to each eye. This can be used to show stereoscopic images.

2.2.2 Large Displays in Real Life

Large displays have been used extensively in a variety of projects and scenarios. While it is beyond the current scope to exhaustively document every project that has ever used a large display, this section highlights some of the work that has explicitly revolved around large displays or that makes interesting use of such displays.

*Engineering a large display*

Nowadays, technology trends and user demands fuel the display industry to produce larger desktop displays for less money. However, for a variety of reasons, high resolution wall-sized displays remain fairly expensive. Hence, the engineering challenge of building these displays out of commodity parts has attracted the attention of several groups. Many of these groups have focused on scalable rendering for large displays (Humphreys and Hanrahan, 1999; Li et al., 2000), creating complex graphics architectures necessary for composing high-resolution images useful in many domains, such as that of data visualization.
Other researchers have focused their efforts on the hardware associated with the actual displays. In desktop computing, researchers have explored the use of multiple display systems, claiming a growing trend of users having multiple monitors associated with their desktop machines. Beyond the desktop, many researchers have worked on combining multiple desktop or projection displays to form large tiled display walls. The PowerWall and InfinityWall (Czernuszenko et al., 1997), as well as the National Computation Science Alliance Display Wall-In-A-Box in Figure 2 fare examples of such systems. A smaller portion of this work has involved less standard display form factors such as curved or domed displays (Raskar, Baar, Willwacher, and Rao, 2004).

**Contextual displays for ambient information**

Because large displays are intrinsically more visible than smaller ones, they can be placed further away or off in the periphery of human vision without making content harder to see or read. Recognizing this capability, researchers have explored the use of large displays to unobtrusively provide contextual information that could be useful to users as they perform their focal tasks on more traditional displays.
In the Prairie system, designed to utilize large displays for distributed knowledge management and collaboration, Swaminathan and Sato identify and support at least four distinct types of contextual information: (1) organizational context, the relationship of a community of users to other communities, (2) social context, the social activities in a community such as presence and current task, (3) work context, how various work objects on the display are related to each other, and (4) navigational context, the path through which a user reaches a particular object (Swaminathan and Sato, 1997).

Baudisch et al provide a large low-resolution overview of the working context around a smaller high-resolution focal screen. In a series of experiments, they showed that the persistent presence of contextual information made users more efficient at tasks that required them to view the focal information at multiple levels of detail (Baudisch, Good, Bellotti, and Schraedley, 2002). MacIntyre et al in their Kimura office environment, assist users to manage multiple working tasks by the system technique’s of presenting interactive montages of images on large peripheral displays such as Figure 3. These montages not only remind users of past actions, but also serve as contextual cues for pending tasks and projects (MacIntyre et al., 2001).

Tan et al build on the principle that the contextual information we incidentally encode when we acquire information in the real world serves as strong memory cues for later
retrieval of that information (Tan, Stefanucci, Proffitt, and Pausch, 2001). In their InfoCockpits system, they utilize large peripheral projection displays to show different scenes of distinct places. These places provide the contexts that serve as cues to help users remember more information. These researchers show a 56% improvement in memory for information presented with the InfoCockpit system as compared to a standard desktop display system. They hypothesize that the greater the sense of presence invoked by the display, the better the memory for learned information.

**Large displays for Public Ad Hoc Social Activity**

In addition to providing ambient information to individuals, researchers have explored placing large displays in key locations such as outside office doors in order to broadcast ambient content that supports social activities in public spaces (Fitton and Cheverst, 2003; Moran et al., 1999; Russell, Trimble, and Wales, 2002). Many of these serve as digital bulletin boards that allow their owners to display text, static images, or rich media content.

As an extension of the bulletin board metaphor, Churchill et al connect multiple large display interactive bulletin boards called Plasma Posters across the network, allowing board owners to post content to multiple locations at once (Churchill, Nelson, Denoue, and Girgensohn, 2003). Similarly, Grasso et al built a large display system called the Community Wall to support information sharing and discovery across communities of practice (Grasso, Roulland, Snowdon, and Muehlenbrock, 2002).

McCarthy et al explore the use of peripheral displays in three workplace contexts: within an individual office, directly outside the office, and in a common area (McCarthy, Costa, and Liongosari, 2001). On the office displays, they present content useful to the individual. Within the other two contexts, they explore the kinds of information that users would like to share with passersby, as well as interaction mechanisms that could aid informal conversations between either local or remote viewers of the displays. While they do not explicitly study the effects of the displays themselves, it is interesting that they choose to use smaller displays for personal viewing and much larger ones as their public displays.
Greenberg and Rounding built Notification Collage, a system that allows users to post text notes and other media and to converse via live video and audio feeds (Greenberg and Rounding, 2001). Using this system, they explored how personal peripheral displays as well as large public displays could enhance casual interaction and communication between users in a community. In their Dynamo system, Izadi et al allow sharing and exchange of information across public displays to support opportunistic meetings in public settings (Izadi et al., 2003). The large display, in this project, serves as a shared digital and physical space on which users can collaborate.

**Large displays in informal group meetings**

Large displays have also been used to support small informal group meetings. In this capacity, large displays have been used as central displays or drawing surfaces that allow easy presentation and capture ideas. Recent work on computer-supported meeting environments has recognized the importance of these central display surfaces. Meeting rooms such as Colab (Stefik et al., 1987), Capture Lab (Mantei, 1988), and Project Nick (Peter et al., 1987) all utilize one or more large displays as a major focus of group work. In fact, Mandryk et al articulate this principle when they identify display size as an important factor in comparing collaborative systems (Mandryk, Scott, and Inkpen, 2002).
The Liveboard system uses a directly interactive, stylus based, large area display to complement other personal computing devices (Elrod et al., 1992). It also provides large shared workspace around which groups can collaborate. Initially, Liveboard simulated standard whiteboard by allowing freehand drawing and erasing. Interestingly, Liveboard-like systems are now commercially available (e.g. SMART Board, see Figure 4).

**Shared Displays for Collaboration**

In addition to facilitating ad hoc social interaction and small informal meetings, large displays have also been used to support more formal cooperative work needed for operating on much larger amounts of information than a single person can handle. In fact, many researchers have built systems that use large public displays to support focused, time critical collaboration. Such systems can already be seen in control rooms of complex real-world systems such as industrial plants, as well as in large planning scenarios such as in military command rooms. For example, the MERBoard system (Trimble, Wales, and Gossweiler, 2002) was designed to help NASA scientists analyze data from Mars rovers.
In their work, Dudfield et al explore the user of panoramic displays to facilitate shared mental models of information, as well as to improve situation awareness and team decision making (Dudfield et al., 2001). In their studies, they found strong user preference for shared large displays, with users reporting improved situation awareness and decision making. However, quantitative analysis of objective data did not support their preference. They hypothesized that this disparity might have been due to the lack of experimental control over the simulation or to insensitivity of objective measures. Other studies that have also concentrated on realistic scenarios in similar military settings support their preference data suggesting that teams do indeed perform better when working on shared large displays (Hiniker, 1998; Hiniker and Entin, 1992).

The Courtyard system was built to support coordination and division of labor by integrating an overview on a shared large display with detailed views on individual personal displays (Tani et al., 1994). Courtyard allows users to access per-user detailed information on their individual screens simply by moving their mouse pointer off their individual screen and pointing to an object on the shared screen.

**Large display environments**

In addition to integrating large displays into more traditional computing or meeting environments, many researchers are creating entirely new computing environments built around large displays. For example, in i-LAND project, Streitz et al. create an environment that supports cooperative work of dynamic teams with rapidly changing needs (Streitz et al., 1999). i-LAND consists of several “roomware” components, or computer-augmented objects that integrate physical elements with digital information technology.

Researchers have used large displays to create immersive rooms for other reasons. For example, Bobick et al. use large displays, coupled with physical objects and digital sensing mechanisms, to create an interactive narrative playspaces for children in their KidsRoom (Bobick et al., 2000).
2.3 Using Large Shared Displays in Interactive Workspaces

Workspaces that are equipped with multiple large shared displays and other interconnected computing devices are known as *interactive workspaces* (Johanson, Fox, and Winograd, 2002). An interactive workspace is a physical space where users interact with personal and shared displays to perform individual or group work. Several distributed infrastructures have been developed that enable multiple, heterogeneous devices to function as a single connected workspace, including Gaia (Román et al., 2002), iRos (Johanson and Fox, 2002), and Aura (Sousa and Garlan, 2002). On top of these infrastructures, many interfaces have been developed to enable seamless redirection of input and relocation of applications (Biehl and Bailey, 2004; Booth, Fisher, Lin, and Argue, 2002; Johanson, Hutchins, Winograd, and Stone, 2002). However, our work introduces guidelines as to when to use large shared displays in collaborative interactive workspaces.

2.3.1 Guidelines for using small/personal and large/shared displays

Before delving into the details of what information should be provided on the large shared displays, the number of large shared displays necessary and their optimal configuration should be studied and determined. Obviously the number of large shared displays depends on factors such as the number of team members and the amount of information that must be displayed. When it comes to large shared displays’ configurations, human factors issues such as visibility and operator satisfaction should be considered before making any final configuration decisions.

Multiple large displays are being increasingly used in interactive workspaces to enhance individual and group work. However, little research has been conducted to determine whether various configurations of large displays impact users differently in terms of their tasks. Su and Bailey show that such an impact exists (Su and Bailey, 2005). Moreover, their research took steps towards developing guidelines for how to effectively arrange large displays in interactive workspaces. Three guidelines were developed: (1) displays can be separated on a horizontal plane up to a subtended visual angle of 45, (2) a display
should not be placed behind a user, but if necessary, it should be offset relative to the user, and (3) displays should be positioned at a 45 angle relative to each other rather than being orthogonal (Su and Bailey, 2005).

**Display Characteristics Important to Large Displays**

There are three important factors that differentiate large displays from smaller ones. These factors are the number of pixels or screen resolution, visual angle or field of view, and physical size. When designers consider large displays, many people think immediately of having more screen space, or more pixels to place more information. This is true of multiple monitor systems, created by adding more displays to traditional single display systems. It is also true of high resolution displays, created specifically to increase screen space and display more information. Another factor of importance is field of view. Large displays are not often placed at a distance that is proportional to their increase in size over small displays. Due to space constraints, they are typically relatively closer and cast a larger retinal image, thus offering a wider field of view.

**Number of Pixels**

Anderson et al studied 108 users working on single monitor and multi-monitor configurations (Anderson, Colvin, and Tobler, 2003). They found that users on multi-monitor setups outperformed users on single monitors on every performance and preference metric they tested. They concluded that adding pixels with multi-monitor setups was cost effective even if tasks they support comprised only about 20 percent of overall work done. Grudin explains how additional pixels provided by multiple monitors can be partitioned to take advantage of focal and peripheral awareness as users work on various tasks (Grudin, 2001). In his paper, he makes several high level observations regarding the use of multiple monitor systems and speculates on how we can design tools to optimally leverage these new systems.
Field of View

When considering field of view, it is important to define precisely what display characteristics are being referred to. There are two field of view angles that must be considered. The display field of view is the angle subtended from the eye to the left and right edges of the display screen. This angle is limited by the physical display width, and can only be increased by replacing the display hardware or moving the user physically closer to the display. The display field of view can be decreased by using a window that is narrower than screen width. The geometric field of view is the horizontal angle subtended from the virtual camera to the left and right sides of the viewing frustum. This angle is under control of the virtual environment designer. Most reported literature does not make the distinction between display and geometric field of views. In most work, the term field of view usually refers to geometric field of view. In this work, field of view is used to refer to the display field of view. However, since we keep a 1:1 correspondence in all this work, the term field of view could just easily refer to the geometric field of view.

It has been reported that it is harmful to deviate from a 1:1 ration of geometric field of view and display field of view (Draper, Viirre, Furness, and Gawron, 2001). Large deviations can cause either magnification or miniaturization of items in the virtual world, possibly leading to discrepancies between studies as well as contributing reliably to simulator sickness. Barfield et al reported that performance was best under mid-sized geometric field of view conditions (45 or 60 degrees) and worst under extreme geometric field of view conditions (30 or 76 degrees), when they had participants judge azimuth and elevation under different conditions of field of view. They concluded that this was because the former geometric fields of view are closest to the display field of view and therefore result in the least amount of distortion.

In summary, it appears that wider fields of view offered by larger displays provide more spatial cues to users and are important aids for many spatial tasks, helping especially with cognitive map construction as the visual complexity of a display or the demands of a task increase.
Physical size is an important cue to sensory and judgment processes in humans. For example, both infants and adults have been shown to exhibit preferences for larger objects in presentations (Silvera, Josephs, and Giesler, 2002). Despite the large amount of work done in comparing fields of view, few researchers have isolated the effects of physical size and distance on the sense of presence or on task performance. In a series of studies, Simmons showed that users performed better on productivity tasks using large 21” monitors as compared to smaller ones (Simmons, 2001). Although this work showed benefits of larger displays, Simmons explored only a small range of display sizes, each viewed at different visual angles and with different resolution. Also, she described the presence of effects without attempting to explain them.

Physically large size and high resolution displays have been widely applied in various fields. Recently Ni et al designed empirical studies to evaluate the individual and combined effects of display size and resolution on task performance in Information-Rich Virtual Environment (IRVE) (Ni, Bowman, and Chen, 2006). Their findings state that users are most effective at performing IRVE search and comparing tasks on large high-resolution displays.

Although previous research investigated various issues related to designing interactive workspaces leveraging the use of large shared displays, we see very little research that gauges the differences between public displays and personal ones in collaborative environments when presenting information in the peripheral level. We believe that there is a gap to be filled between designing interfaces on large displays in collaborative environments and designing group notification systems. Therefore, we intend to study this gap and provide tradeoffs and guidelines for designers to maximize individual and group performances in various workspaces.
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 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 5.
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.

3.2.3 Participants

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.
3.2.4 Materials

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 6.

![Diagram](image)

Figure 6: The large shared display is a projected display in the same field of view as the personal display.

3.2.5 Tasks

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.

3.2.6 Procedure

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.

3.2.7 Results

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.

![Experiment 1: Average Words Per 3 Minutes](image)

Figure 7: 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 7. 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.

3.2.8 Summary

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 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.
3.3 Experiment 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.

3.3.1 Hypotheses

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.

3.3.2 Experimental 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.

3.3.3 Participants

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 2.7 (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.

3.3.4 Tasks

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.
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 8. Although Augur provides line-oriented view like SeeSoft (Eick, Steffen, and Eric E. Sumner, 1992), it combines activity information as one of the visualization attributes. Augur provides a per-line level view of activity, good for learning specific changes within a single or small set of code files.

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.
3.3.5 Procedure

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.

3.3.7 Summary

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.

3.4 Conclusion

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.
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 peripheral 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.
4.2 Source 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, and Eric E. Sumner, 1992), a line oriented visualization tool, and was enhanced by several tools. Among those tools are Augur (Froehlich and Dourish, 2004), Aspect Browser (Griswold, Yuan, and Kato, 2001), sv3D (Marcus, Feng, and Maletic, 2003), Tarantula (Jones, Harrold, and Stasko, 2002), Gammatella (Orso, Jones, Harrold, and Stasko, 2004), and ALMOST (Renieris and Reiss, 1999).

Augur is a visualization tool that allows distributed software development. It creates visualizations of software artifacts, software development activities, and relationships among authors. Aspect Browser is a tool developed using the map metaphor to track software evolution. The sv3D system used a 3D representation for the lines of code in order to compact the screen space needed. Tarantula was developed primarily to aid the debugging process, by offering visualizations to help the debugger locate faults. Gammatella offers three levels of visualization, statement level, file level, and system level. At the statement level, the source code itself is represented. At the file level, a pixel-line view of the source code is created. The system level uses treemaps to represent the software. At each level, coloring is used to represent information about the code.

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, Voinea, and Telea, 2005) and Augur (Froehlich and Dourish, 2004).

### 4.3 SYSTEM 1: VISUAL CODE NAVIGATOR (VCN)

The VCN system 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 $V_i$ for every file $F_j$, for every version, attributes such as creation date and author are also recorded.

#### 4.3.2 Views 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 (Voinea, Telea, and van Wijk, 2004; Voinea, Telea, and Wijk, 2005; Wijk and Wetering, 1999). 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 9.

Figure 9: VCN Syntactic View

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 10.

![Figure 10: VCN Symbol View](image)

(3) Evolution View

This view, shown in Figure 11, 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.
Augur 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.

4.4.1 Implementation

Augur requires no additional setup to run; however, it uses the information already available in configuration management systems.
4.4.2 Views 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, and Eric E. Sumner, 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 12.

![Augur Overview](image)

Figure 12: Augur Overview

(2) HISTORY VIEW

Figure 13 shows the history view in the Augur interface. In the history view, lines are colored according to their change history.
Similarly, the author view in Figure 14 shows the authors involved in the development of a certain file.

Figure 13: Augur History View

(3) AUTHOR VIEW

Similarly, the author view in Figure 14 shows the authors involved in the development of a certain file.
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.

4.5 Experiment 3

4.5.1 Hypotheses

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.

4.5.2 Experimental 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.
4.5.3 Participants

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 2.7 (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.

4.5.4 Procedures

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.

4.5.5 Results

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.

4.5.6 Summary

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.

4.6 Interviews

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.

4.6.1 Results

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.

**4.7 Guidelines**

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.
4.7.2 Guideline 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 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.

4.8 Conclusion

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.
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 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 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 (Berntsson-Svensson and Aurum, 2006; Boonzaaier and Loggerenberg, 2006; Nienaber and Cloete, 2003; Padayachee, 2002; Wongthongtham, Chang, and Dillon, 2004), 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.

In their field study, Curtis et al (Curtis, Krasner, and Iscoe, 1988) investigated how design decisions were made, represented, communicated and updated throughout the development process. They found that teams face significant communication and coordination breakdowns attributed to the absence of appropriate communication tools that supported their work practices. Similarly, Karut and Streeter’s analysis showed that programmers lack effective informal communication tools, especially those that promote a team-wide view of a project’s organization (Kraut and Streeter, 1995).

More broadly, several studies have investigated the overall breakdown of a programmer’s time. These studies report that maintaining an awareness of a project’s status consumes a significant proportion of a programmer’s time (LaToza, Venolia, and DeLine, 2006; Linebarger, Scholand, Ehlen, and Procopio, 2005). One study reports programmers spend as much as 40% of their work day communicating about code (LaToza, Venolia, and DeLine, 2006).

In this chapter, we 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).
5.2 Exploring Group Awareness Needs

Group awareness, the understanding of who you are working with, what is being worked on, and how your actions affect others, is essential to effective collaboration (Dourish and Bellotti, 1992). 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. A common method for gaining and maintaining this awareness is through the use of shared artifacts (Tang, 1991). In their study of group awareness needs in various types of software development environments, Biehl et al found that programmers use a key set of information items to form an awareness of others’ activities, and that programmers see great potential for large shared displays to be used for awareness, but their existing tools were not effective for this use (Biehl, Czerwinski, Smith, and Robertson, 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 HTA was built, as shown in Figure 15, to provide a means of understanding the current processes and activities undertaken to develop software.
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.

5.3 Designing 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 16, 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 16 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.

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 17.
Designing to Enhance Group Awareness

5.3.2 Project 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

Figure 17: Time tracking view showing that member M3 finished all his work and is considered an empty resource that should be utilized
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 18. 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 18: Project tracking view showing the progress of each activity along with the members assigned to work in each activity](image)
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.

5.3.3 Dependencies 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. Moreover, some dependencies depend on each others such as internal dependencies depending on each others 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.

![Gantt-Chart]

Figure 19: 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 20. 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 20 should provide insight in an in-situ study as to possible benefits for coordinated group information on a shared display.

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 and testers. 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>
<thead>
<tr>
<th>Number of participants</th>
<th>Role of participant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project manager</td>
</tr>
<tr>
<td>1</td>
<td>Team leader</td>
</tr>
<tr>
<td>6</td>
<td>Junior JAVA developers</td>
</tr>
<tr>
<td>4</td>
<td>Senior JAVA developers</td>
</tr>
</tbody>
</table>

Table 3: Participants roles in the in-Situ study

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 21. Again we followed the environment setup mentioned in the guidelines for placing large displays (Su and Bailey, 2005).
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.

5.4.6 Results 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
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 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, and Kuchinsky, 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”.
5.4.7 Summary

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.

5.5 Conclusion

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.
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.

### 6.1 Summary 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.

6.2 Major 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.

6.3 Future 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
7.1 Appendix A

7.1.1 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?
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
11. What was the name of the café?

- [ ] Cilantro
- [ ] Carlos
- [ ] Costa

12. How many laptops were there?

- [ ] 1
- [ ] 2
- [ ] 3

7.1.2 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?
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
### 7.2.1 First experiment results

<table>
<thead>
<tr>
<th>Participants</th>
<th>Base (WP3Mins)</th>
<th>Personal Display (WP3Mins)</th>
<th>Large Display (WP3Mins)</th>
<th>Personal Display (Questions)</th>
<th>Large Display (Questions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>157</td>
<td>145</td>
<td>140</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>210</td>
<td>198</td>
<td>195</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>57</td>
<td>55</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>84</td>
<td>79</td>
<td>72</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>200</td>
<td>186</td>
<td>183</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>115</td>
<td>90</td>
<td>100</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>139</td>
<td>130</td>
<td>125</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>82</td>
<td>69</td>
<td>67</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>70</td>
<td>50</td>
<td>55</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>138</td>
<td>126</td>
<td>124</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>108</td>
<td>94</td>
<td>92</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>224</td>
<td>219</td>
<td>200</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>13</td>
<td>136</td>
<td>129</td>
<td>123</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>155</td>
<td>143</td>
<td>140</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td></td>
<td>221</td>
<td>211</td>
<td>209</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>171</td>
<td>165</td>
<td>154</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>112</td>
<td>98</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>107</td>
<td>96</td>
<td>95</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>214</td>
<td>162</td>
<td>168</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>231</td>
<td>195</td>
<td>198</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>109</td>
<td>85</td>
<td>83</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>206</td>
<td>191</td>
<td>189</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>142</td>
<td>125</td>
<td>120</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>137</td>
<td>132</td>
<td>124</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>191</td>
<td>187</td>
<td>180</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>153</td>
<td>138</td>
<td>132</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>73</td>
<td>60</td>
<td>54</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>196</td>
<td>182</td>
<td>176</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>56</td>
<td>50</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>93</td>
<td>80</td>
<td>75</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>
### 7.2.2 Second experiment results

<table>
<thead>
<tr>
<th>Participants</th>
<th>Base Condition (Time Sec)</th>
<th>Personal (Primary Time Sec)</th>
<th>Public (Primary Time Sec)</th>
<th>Personal Questions (Time Sec)</th>
<th>Public Questions (Time Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.59</td>
<td>10.79</td>
<td>11.90</td>
<td>2.23</td>
<td>1.65</td>
</tr>
<tr>
<td>2</td>
<td>10.78</td>
<td>11.75</td>
<td>10.37</td>
<td>2.54</td>
<td>2.03</td>
</tr>
<tr>
<td>3</td>
<td>10.17</td>
<td>11.87</td>
<td>11.11</td>
<td>3.20</td>
<td>2.36</td>
</tr>
<tr>
<td>4</td>
<td>8.94</td>
<td>11.29</td>
<td>10.16</td>
<td>1.98</td>
<td>1.96</td>
</tr>
<tr>
<td>5</td>
<td>9.02</td>
<td>11.08</td>
<td>9.17</td>
<td>2.54</td>
<td>2.95</td>
</tr>
<tr>
<td>6</td>
<td>11.85</td>
<td>9.31</td>
<td>10.00</td>
<td>3.14</td>
<td>2.49</td>
</tr>
<tr>
<td>7</td>
<td>9.13</td>
<td>10.89</td>
<td>8.20</td>
<td>2.23</td>
<td>2.06</td>
</tr>
<tr>
<td>8</td>
<td>9.85</td>
<td>9.41</td>
<td>9.28</td>
<td>2.53</td>
<td>2.58</td>
</tr>
<tr>
<td>9</td>
<td>11.28</td>
<td>11.03</td>
<td>8.98</td>
<td>2.28</td>
<td>2.44</td>
</tr>
<tr>
<td>10</td>
<td>10.63</td>
<td>10.69</td>
<td>9.97</td>
<td>2.46</td>
<td>1.99</td>
</tr>
<tr>
<td>11</td>
<td>10.33</td>
<td>10.44</td>
<td>11.39</td>
<td>2.17</td>
<td>2.41</td>
</tr>
<tr>
<td>12</td>
<td>11.14</td>
<td>10.16</td>
<td>9.67</td>
<td>2.56</td>
<td>1.68</td>
</tr>
<tr>
<td>13</td>
<td>8.49</td>
<td>8.82</td>
<td>8.26</td>
<td>2.26</td>
<td>2.55</td>
</tr>
<tr>
<td>14</td>
<td>9.84</td>
<td>9.08</td>
<td>10.08</td>
<td>2.27</td>
<td>1.68</td>
</tr>
<tr>
<td>15</td>
<td>10.34</td>
<td>10.20</td>
<td>9.70</td>
<td>3.46</td>
<td>2.08</td>
</tr>
<tr>
<td>16</td>
<td>9.36</td>
<td>9.35</td>
<td>11.22</td>
<td>1.98</td>
<td>1.58</td>
</tr>
<tr>
<td>17</td>
<td>8.27</td>
<td>11.57</td>
<td>10.37</td>
<td>1.66</td>
<td>1.32</td>
</tr>
<tr>
<td>18</td>
<td>10.92</td>
<td>11.38</td>
<td>8.68</td>
<td>1.78</td>
<td>2.23</td>
</tr>
<tr>
<td>19</td>
<td>9.17</td>
<td>11.53</td>
<td>10.52</td>
<td>2.38</td>
<td>2.28</td>
</tr>
<tr>
<td>20</td>
<td>9.67</td>
<td>9.27</td>
<td>11.16</td>
<td>2.76</td>
<td>2.22</td>
</tr>
<tr>
<td>21</td>
<td>10.59</td>
<td>10.78</td>
<td>9.50</td>
<td>1.97</td>
<td>1.24</td>
</tr>
<tr>
<td>22</td>
<td>9.09</td>
<td>11.18</td>
<td>11.47</td>
<td>1.84</td>
<td>1.71</td>
</tr>
<tr>
<td>23</td>
<td>10.40</td>
<td>11.63</td>
<td>11.24</td>
<td>2.52</td>
<td>2.40</td>
</tr>
<tr>
<td>24</td>
<td>8.39</td>
<td>11.70</td>
<td>8.68</td>
<td>2.41</td>
<td>1.72</td>
</tr>
<tr>
<td>25</td>
<td>9.30</td>
<td>11.52</td>
<td>8.36</td>
<td>2.69</td>
<td>1.70</td>
</tr>
<tr>
<td>26</td>
<td>10.65</td>
<td>8.09</td>
<td>11.61</td>
<td>1.90</td>
<td>2.14</td>
</tr>
<tr>
<td>27</td>
<td>10.88</td>
<td>11.84</td>
<td>9.05</td>
<td>2.75</td>
<td>1.52</td>
</tr>
<tr>
<td>28</td>
<td>11.99</td>
<td>10.53</td>
<td>11.09</td>
<td>2.64</td>
<td>2.20</td>
</tr>
<tr>
<td>29</td>
<td>8.00</td>
<td>8.88</td>
<td>8.97</td>
<td>2.18</td>
<td>1.99</td>
</tr>
<tr>
<td>30</td>
<td>9.94</td>
<td>8.59</td>
<td>9.72</td>
<td>3.17</td>
<td>1.22</td>
</tr>
</tbody>
</table>
REFERENCES


Vita

Khaled Hussein

Education
1. June 2007 | M.S. Computer Science and Applications (Virginia Polytechnic Institute and State University)
   - “Towards the Development of User Interface Guidelines for Large Shared Displays”, Scott McCrickard (Advisor), Chris North (Committee Member), Ayman Abdel-Hamid (Committee Member)
   - GPA = 3.7/4.0

2. June 2005 | B.S. Computer Science (Arab Academy for Science and Technology and Maritime Transport)
   - GPA = 4.0/4.0

Experience
1. March 2005 – December 2005 | Project Manager
   Lotus Tech Co.
   Managed two projects implemented using .NET Framework and JAVA, participated in analysis, and provided my recommendations that raised profit of the company to 20% instead of 6%.

   iSuperHosting
   Developed E-Business Solutions, XML Web Services, Web Hosting Panels, and Administration tools.

   StarMind LLC
   Developed e-Business Solutions and ecommerce applications with ASP.NET and JSP. Moreover, I customized enterprise resource planning applications. Development was in .NET Framework.

Technical Experience

Programming Languages
- C#, C/C++, Java, Visual Basic.NET.
- Ruby on Rails, Lua, Scheme, Prolog.

Tools and Techniques
- Win Forms, .NET Compact Framework, Web Services.
- Graphics Programming: Managed DirectX 9.0c, GDI, GDI+
- Data Access: JDBC, ODBC, OLEDB, ADO.NET.
Vita

- DBMS: SQLServer, Oracle, MySql Server.
- Encoders: Windows Media Encoder, Video Compression, broadcasting, Video Streaming.

Sample Projects

**E Business Solutions**

1) **E University: Virtual Online University.**

My responsibilities include leading the team, manage the project progress, and development in which an online adaptive test engine was developed for professors to design and author online exams. This enables students to take exams and get graded automatically. Moreover, a broadcasting module was implemented using Microsoft Media Encoding SDK to facilitate application level multicasting for students to attend lectures with high quality video formats.

2) **E-mail proxy.**

It is a proxy server that provides extra functionalities like encrypting and decrypting the e-mails that are coming out of the internal network to external network. This application also provides the encryption/decryption functionality to any file that is being transferred in/out the internal network. It was implemented using Java.

**Enterprise Applications**

1) **Real Estate Management System.**

It is an enterprise application for a real estate company in USA. This application is responsible for having all the items ready to be showed to their customers. Moreover, the system suggests the best item for the customer needs.

2) **Letter Engine**

Fully utilized system for processing script oriented business letters. The system provides scripting language in which letters are written. The letter gets parsed, compiled, and assembled at run time. Finally, the engine generates a customer oriented letter.

**Scientific Projects**

1) **SuperTrack: Tracking Projects in Collaborative Software Development Environments**

Developed a system for tracking software projects in collaborative software development environments in which a large shared display is used. A set of usability tests was conducted in IBM Egypt. Moreover, an In-Situ study has been conducted to test the various types of effects of the system in RiadaSoft Co.

2) **Electronic Digital Circuits Designer.**

It enables digital circuits’ designers to design electronic circuits using basic gates (AND, OR, INV) and simulate the resulted circuit using LEDs.

3) **DSS Academic Advisor.**
Vita

A Decision Support System used for providing the best solutions regarding academic advising, allowing more free time for academic advisors to look into special cases.

4) Automatic Scheduler
A Software System for automating the generation of university timetables and schedules within hard and soft constraints with extensible architecture.

5) Implementation of GJ algorithm on parallel cluster
It was required to implement Gauss Jordan algorithm for solving a system of linear equations using C and LAM/MPI on a cluster of 124 processors. I was able to achieve the optimal implementation of solving 5000 equation with 5000 unknowns in 30 Sec.

Languages
- Arabic: Native language
- English: Write, read, and speak fluently
- German: basic skills