



Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	The Design and Evaluation of Graphical Radial Menus
Author(s)	Samp, Krystian
Publication Date	2011-12-05
Item record	<a href="http://hdl.handle.net/10379/2672">http://hdl.handle.net/10379/2672</a>

Downloaded 2024-04-26T20:16:45Z

Some rights reserved. For more information, please see the item record link above.



Digital Enterprise Research Institute

National University of Ireland, Galway

# **THE DESIGN AND EVALUATION OF GRAPHICAL RADIAL MENUS**

Krystian Samp

under the supervision of Prof. Stefan Decker  
at the Digital Enterprise Research Institute

A thesis submitted in conformity with the requirements  
of the Degree of Doctor of Philosophy

December 2011

Copyright © 2011 Krystian Samp



To my parents, Kazimierz and Krystyna,  
my sisters, Kasia, Ania, and Magda,  
and my love Yolanda



## Acknowledgments

I would like to thank my parents, Kazimierz and Krystyna, for their endless love, support, advice and patience in instilling the values which I am so happy to carry in me today. Thanks to my sisters for their love. Thanks to my love Yolanda who day to day supported me in everything. I am grateful to Bill McDaniel, who let me fully develop my HCI interests and provided me with great opportunities. I am indebted to my supervisor, Prof. Stefan Decker, who allowed me to take on an HCI topic and supplied me with enthusiasm and creative insight. I am also thankful to Prof. Alan Dix who agreed to be my external examiner. I was also fortunate to be here, in Galway, the best city in the world!



## Abstract

Graphical menus provide slow performance and yet users often try to select menu commands as fast as possible. Shortcut techniques that enable users to by-pass selections from a graphical menu are fast, but are not the ultimate answer to the performance problem. Shortcut techniques pose high demands on human memory. They require much training and frequent use. They do not scale well. And they often cannot be provided on devices with limited capabilities, such as mobile devices with small screens, small keyboards, or touch screens. Because of these problems, users often times 'stick' to a graphical menu and are condemned to its poor performance.

This thesis argues that the performance of graphical menus can be improved. The thesis consists of three parts: The first part presents the background knowledge and the problem of slow graphical menus. The second part presents the design of a menu that supports the needs of novices, experts, and those transitioning from novice to expert. In particular, the design explicitly aims at improving selection performance by leveraging concentric radial layout. The third part of the thesis presents the evaluation of the design and its main characteristics. Four studies are conducted based on two controlled, user experiments. A number of variables are analysed including pointing time, visual search time, navigation performance, novice/expert performance, navigation errors, and screen consumption. A number of factors are controlled including menu size, location of menu items, menu design, and layout variation. Traditional linear menus are included for comparison. Objective and subjective measures are used. Additionally, a performance model based on pointing laws is proposed and used to measure the motor processing rates of tested designs.

The evaluation shows that designing with performance in mind can result in faster graphical menus compared to traditional menus. The proposed Compact Radial Layout menu is approximately 20-30% faster in a variety of conditions, reduces navigation errors, induces users to faster movements and decreases screen consumption. The evaluation also shows that, with a consistent pattern of organizing items in radial menus, search and navigation performance can be further improved. At the same time, the results indicate that the improved navigation performance in radial menus comes at the cost of slower visual search.



## Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Motivation .....	1
1.2. Goals .....	2
1.3. Outline .....	2
<b>2. Background (historical perspective)</b>	<b>5</b>
2.1. Evolution of computer use .....	5
2.2. Human-Computer Interface.....	6
2.3. Novices and Experts.....	7
2.4. Supporting Novices: Graphical User Interfaces .....	8
2.5. Supporting Experts: dedicated techniques .....	10
2.6. Separation consequences .....	13
2.7. Potential benefits of faster menus.....	19
2.8. Related work .....	21
<b>3. Design</b>	<b>29</b>
3.1. Design space.....	29
3.2. Design goals.....	29
3.3. Design characteristics .....	34
3.4. Design description .....	56
3.5. Summary of the design .....	71
3.6. Roadmap for experiments.....	73
<b>4. Speed/Accuracy Trade-Off in Hierarchical Menu Navigation</b>	<b>75</b>
4.1. Background.....	76
4.2. Sequential Pointing Model .....	87
4.3. Experiment.....	90
4.4. Menu comparison .....	94
4.5. Modelling.....	98
4.6. Discussion.....	104
4.7. Conclusions.....	105
<b>5. Navigation Time Variability: Measuring Menu Navigation Errors</b>	<b>107</b>
5.1. Related Research .....	108
5.2. Navigation Time Variability measure .....	110
5.3. Experiment .....	112
5.4. Conclusions.....	114

<b>6. Supporting Menu Design with Radial Layouts</b>	<b>117</b>
6.1. Experiment.....	117
6.2. Results.....	122
6.3. Discussion.....	127
6.4. Conclusions.....	131
<b>7. Visual Search and Mouse-Pointing in Radial Menus</b>	<b>133</b>
7.1. Method.....	133
7.2. Results.....	135
7.3. Conclusions.....	142
<b>8. Conclusions and Future Work</b>	<b>143</b>
8.1. CRL menu.....	143
8.2. Radial layout in Graphical User Interfaces.....	146
8.3. Outcomes.....	148
<b>References</b>	<b>151</b>

## ***1. Introduction***

### ***1.1. Motivation***

The last decades have been marked with the explosive growth of computing. Hardware is becoming smaller, faster, parallel, and functionally more capable. Software continues to turn hardware into powerful platforms, providing ever more advanced functionality at the fingertips of programmers. And new product designs blend hardware and software into new areas of human lives.

The growth of computing has certainly not reached its end. The Semantic Web and the concept of Networked Knowledge challenge the frontier of information meaning and accessibility. Computers are becoming capable of ‘understanding’ information in great detail and on a Web scale. This is particularly important when it comes to enabling new scenarios where computers support humans. Especially, supporting human information needs that underlie so many everyday activities.

The growth of computing means that people spend more and more time communicating with computers. This communication takes place at the human-computer interface where people translate their thoughts into computer representations, and computers translate data into representations that can be comprehended by people. The effectiveness of this communication directly influences values such as productivity, creativity, and satisfaction. And any problems at the human-computer interface detract from these values.

This thesis focuses on one of the problems of human-computer interface, namely, the problem of slow performance of graphical menus. A current approach for dealing with this problem is to provide supplementary techniques, shortcuts for by-passing selections from a graphical menu. Examples of shortcuts include keyboard accelerators, gestures, and vocal commands. It turns out, however, that the utility of shortcut techniques is limited. This limitation stems predominantly from the fact that shortcut techniques heavily rely on recall; that is, they pose high demands on human memory. They require much training and frequent use. They also do not scale well and

often cannot be provided in computing contexts employing devices with limited capabilities, such as mobile devices with small screens and small keyboards.

The limited utility of shortcut techniques mean that users often times ‘stick’ to a graphical menu instead of transitioning to shortcut techniques. In these situations, users are condemned to the poor performance of graphical menus; poor, because they were not designed with performance in mind. This situation impacts the effectiveness of human-computer communication. And the scale of this impact is stressed by the fact that menus are one of the most ubiquitous and widely used interface techniques employed in the majority of applications.

To remedy this problem, graphical menus need to provide better performance. And this thesis presents the argument that such menus can be created.

### **1.2. Goals**

The central hypothesis of this work is that graphical menus can be designed with performance in mind, and that such graphical menus can perform better than the most widely used cascading menus. Improved performance has many dimensions including shortened selection times, decreased navigation difficulty, lowered error rates, and inducing users to fast movements that require little precision.

To prove this central hypothesis the thesis focuses on achieving two subgoals. The first subgoal is to build an informed graphical menu design that explicitly includes the requirement of effective and efficient selections. The second subgoal is to conduct a thorough evaluation of the proposed design and its characteristics.

### **1.3. Outline**

The thesis consists of 7 chapters which are summarized below:

- Chapter 2 begins by providing a historical perspective of graphical menus and shortcut techniques for faster menu selections. Problems of such interface configuration are discussed. It is shown that graphical menus should be designed with performance in mind. The related work on improving menu performance is reviewed.
- Chapter 3 presents a graphical menu design that supports users of different expertise levels. In particular, the design aims at enabling effective and efficient selections required by the most experienced users. The presentation includes the

design goals, the design space, the menu characteristics directly affecting our goals, and the description of the final design.

The following chapters evaluate the proposed menu design and its key characteristics:

- Chapter 4 presents a controlled experiment that compares navigation performance of the proposed menu and a cascading menu using both qualitative and quantitative measures. Apart from the comparison, the chapter also proposes a model of navigation which is based on pointing laws. This model is used with the collected data to deepen the insights concerning the navigation characteristics of both menu designs.
- Chapter 5 compares the navigation errors from the proposed menu and a cascading menu. The chapter begins with the observation that the current approach of measuring selection errors is limited to only one specific type of errors: selections missing a target item. A new measure of navigation errors is proposed; it captures the severity of all navigation errors. Then, the current approach and this new measure are used to analyse the data collected in the experiment presented in Chapter 4.
- Chapter 6 presents a controlled experiment that compares the visual search time and pointing time of linear and radial layout. These layouts are the basis of both existing cascading menus and the proposed menu. The experimental conditions include two menu sizes, three menu levels, and four layouts (one linear and three radial). Both qualitative and quantitative measures are used. Apart from comparing visual search and pointing, novice and expert performance are assessed. The screen consumption is also compared.
- Chapter 7 analyses how items are searched for and pointed-at in menus using radial layout. The objective of this investigation is to help order menu items in a way that improves both search and pointing performance. This chapter uses the data from the experiment presented in Chapter 6.
- Chapter 8 presents the conclusions.



## ***2. Background (historical perspective)***

### ***2.1. Evolution of computer use***

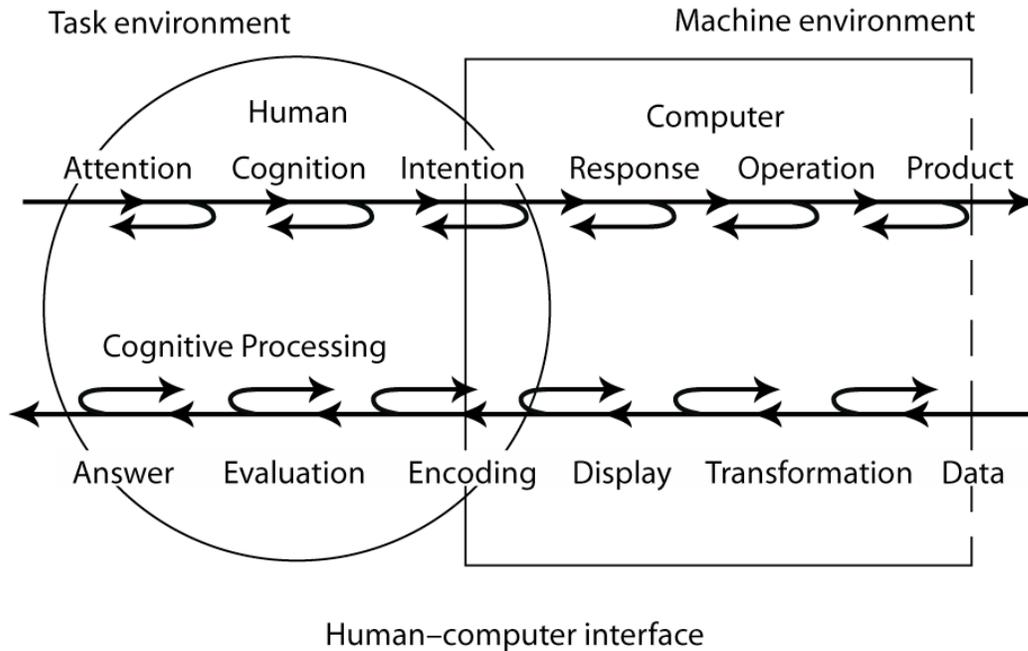
Back 30 years ago, computers required technical staff and were intended to be used by experts who knew programming languages and understood the underlying principles of computer operation. A user carefully crafted a program on a sheet of paper and passed it to a keypunch operator, who used an offline device to create a deck of punched cards that encoded the same program in a different form. The cards were further passed to a computer operator, who entered the program into the computer using a card reader. The computer responded with printed messages, and these were sent back to the user for further inspection.

As computers started to become smaller, more powerful and cheaper a new aspiration was born: for computers to break free from university laboratories, enter offices and homes, and serve a broad range of users for a broad range of tasks [Engelbart, 1968]. Computers were sought to *augment man's intellect* [Engelbart, 1962]. This aspiration was embodied in the development of personal computers. An important step towards realizing this aspiration was made by changing the scheme of entering information into the computer. Namely, human intermediaries were removed. The user and the computer were coupled in a direct conversation at the human-computer interface.

It was clear that the quality and the effectiveness of the interface would be among the deciding factors of the success of the computers. Consequently, it became increasingly important to understand and characterize the processes pertaining to the human-computer interface in great detail. Commensurate with this importance, in 1983, a book by Card, Moran, and Newell established a new field of study called Human-Computer Interaction.

## 2.2. Human-Computer Interface

It has been proposed to consider a human-computer interface in terms of the flow and feedback of information through that interface [Norman et. al., 1980; Norman, 1991a; Norman, 1991b]. This is schematically depicted in Figure 2.1.



**Figure 2.1. Human-Computer Interface.** The information and control flows between human and computer. This is an original figure from [Norman, 1991b, 2008]. Courtesy of Kent L. Norman.

According to this model, a user needs information from the computer, and the computer cannot function without information from the user.

A main component is *the flow of control*. This is a sequence of steps in a process that is determined by a set of rules. Within the human, the flow of control is a sequence of mental operations determined by cognitive processes. Within the computer, the flow of control is determined by a program residing in memory in conjunction with the processor.

Both, the human (represented by a circle) and the machine (represented by a rectangle) reside in environments that provide information, constraints, and contexts. The non-overlapping area within the circle represents cognitive processes involved in the tasks that are not directly related to the human-computer interface. Similarly, the

non-overlapping area within the rectangle represents the computer procedures involved in the tasks that are not directly related to the interface. The overlapping area represents processes that pertain to the interface. From the user side, these include the mapping of information to machine a representation of data through keying or other input devices. From the computer side, these processes include mapping machine representations of data to information presented on the screen or other output devices. Here, at the human-computer interface (the overlapping area), the flow of control is shared and at times passed back and forth between the user and the machine. The computer gives information prompting the user for input, and the user supplies input that directs the subsequent operations.

It is important to note that according to the model, the interface is an area. This captures an idea that the interface is not merely a thin line through which information travels, but rather it is a shared surface that includes the user's cognitive model of the system and the system's model of the user.

### ***2.3. Novices and Experts***

The human-computer interface allowed users to directly communicate with computers. However, the first personal computers required a fair amount of expertise. The interfaces were textual and required knowledge of the underlying programming language, such as BASIC, or a command language, such as those provided by UNIX systems. Such interfaces were only well-suited to existing computer experts or new users with technical skillsets who had the motivation to spend their time reading manuals and learning details. But the aspiration of personal computers was to reach ordinary people who often did not have knowledge regarding computers, technical skills, or time for long training.

In response, some early approaches focused on providing “minimal manuals” [Carroll, 1984, 1990; Wendel and Frese, 1987]. The studies demonstrated, however, that new users did not read manuals regardless of how simple or minimal they were. Instead, they attempted to use software immediately [Carroll and Mazur, 1985; Carroll et. al., 1985; Carroll and Rosson, 1987; Wright, 1988; Cuthbert, 1992; Lazonder, 1994], even though they could reduce confusion and improve long-time performance by taking some initial time for training (the so called *paradox of the active user*). The actual behaviour of new users seemed to be far from the assumed behaviour of idealized rationals.

At that point in time it became clear that for the interfaces to be effective, they should not support any assumed behaviour, but rather the actual user behaviour and needs. Consequently, research studies began to explore the characteristics of user behaviour with computers. Quickly notable differences were found between less and more experienced users. The differences related to cognitive processes involved on the human side of the information flow.

The HCI field developed a notion of the user *expertise continuum* [Preece and Rogers, 1994] which differentiated users according to two overlapping criteria: 1) knowledge and 2) the time spent using a system and/or computer [Prümper et. al., 1991; Nielsen, 1993; Preece and Rogers, 1994; Shneiderman and Plaisant, 2004]. According to this notion, on one end of the continuum there are *novices*. They encounter a task, computer, or interface for the first time or use it infrequently. They plan and think what to do next. But as mentioned before, they do not want to invest their time in extensive training. Novices' efficiency is limited by cognitive load [Padilla, 2003]. On the other end of the continuum, there are *experts*. They are familiar with the task and they use the computer and the interface frequently. In contrast to novices, experts do not exhibit decision-making behaviour. They know the interface elements and how to use them. Experts program specific interactions and perform tasks automatically, often without consciously having to think about them [Welford, 1968; Card et. al., 1983; Norman, 1991b]. Experts' efficiency is limited by their motor load [Padilla, 2003].

The considerable differences in novice and expert behaviour called for different support from human-computer interfaces. Novices needed simplicity, guidance, and an easy way to learn how to use an application. Experts, on the other hand, needed efficiency and full control over the system.

## **2.4. Supporting Novices: Graphical User Interfaces**

A major advancement in the direction of simplifying computer use and supporting the actual behaviour of novices was the advent of *Graphical User Interfaces* (GUI). Three important aspects contribute to the success of GUIs.

### **2.4.1. Reliance on recognition**

GUIs present the application functionality and state *visually*. Therefore, a user does not have to start by reading the documentation but rather can look at the application,

try to find relevant interface elements, use them, and eventually learn by trial and error. GUIs can take advantage of affordances to make it clear how they work just by looking at them [Norman, 1988, 1999; Gaver, 1991]. For example, a button looks like something that can be pressed to activate a function represented by the button label.

### **2.4.2. Familiar metaphor**

GUIs adopted a familiar mental model, the desktop metaphor and its careful extension in the **Windows, Icons, Menus, and Pointing** interface (WIMP). Fredrick Brooks aptly presents the strength of this metaphor with the following words:

*“Dragging and dropping follow directly from the metaphor; selecting icons by pointing with a cursor is a direct analogy of picking things with the hand. Icons and nested folders are faithful analogies of desktop documents; so is the trash can. The concepts of cutting, copying, and pasting faithfully mirror the things we used to do with documents on desktops. So faithfully is the metaphor followed and so consistent is its extension that new users are positively jarred by the notion of dragging a diskette’s icon to the trash to eject the disk”*

Brooks [1995], p.261

The above analogies made WIMP interfaces easy to introduce to novices. They also further reduced learning times and encouraged a trial-and-error approach. The WIMP concept was first publicly presented in 1968 by Doug Englebart and his team from the Stanford Research Institute [Engelbart, 1968]. From there the ideas went to Xerox Palo Alto Research Center, Apple Inc. and Microsoft.

The most significant extension of the desktop metaphor introduced by the WIMP interfaces was a graphical menu. Such a menu displays a list of commands and provides a mechanism by which users can choose one of these commands. While graphical menus display commands in substantial detail, they require only a minimal effort to select them [Lee and Raymond, 1993]. Such a mode of interaction greatly simplified the language of human-computer communication to simple, unitary actions.

### **2.4.3. Direct manipulation**

GUIs, including WIMP, adopted a direct manipulation technique [Shneiderman, 1983] to interact with application content and application interface elements, such as buttons, tabs, menus, scrollbars, and links. In the case of an application interface, a user points at a desired interface element and selects it by pressing a button. The same simple technique is used regardless of the functional complexity ‘hidden’ behind a

manipulated interface element. Pointing devices, such as a mouse or a stylus, made the direct manipulation technique an incredibly easy and natural task.

From the view of human-computer interface discussed in Section 2.2, GUIs simplified and restricted input from the user but provided much in the way of prompting the user. The restriction of the capabilities mixed with rich visual aids made GUIs successful in bringing computer to a broad range of new users, i.e., novices. Several studies demonstrated that software with GUIs is easy to learn and use, and often result in superior task completion times [Temple and Sloane, 1990; Davis and Bostrom, 1992; Galitz, 1996]

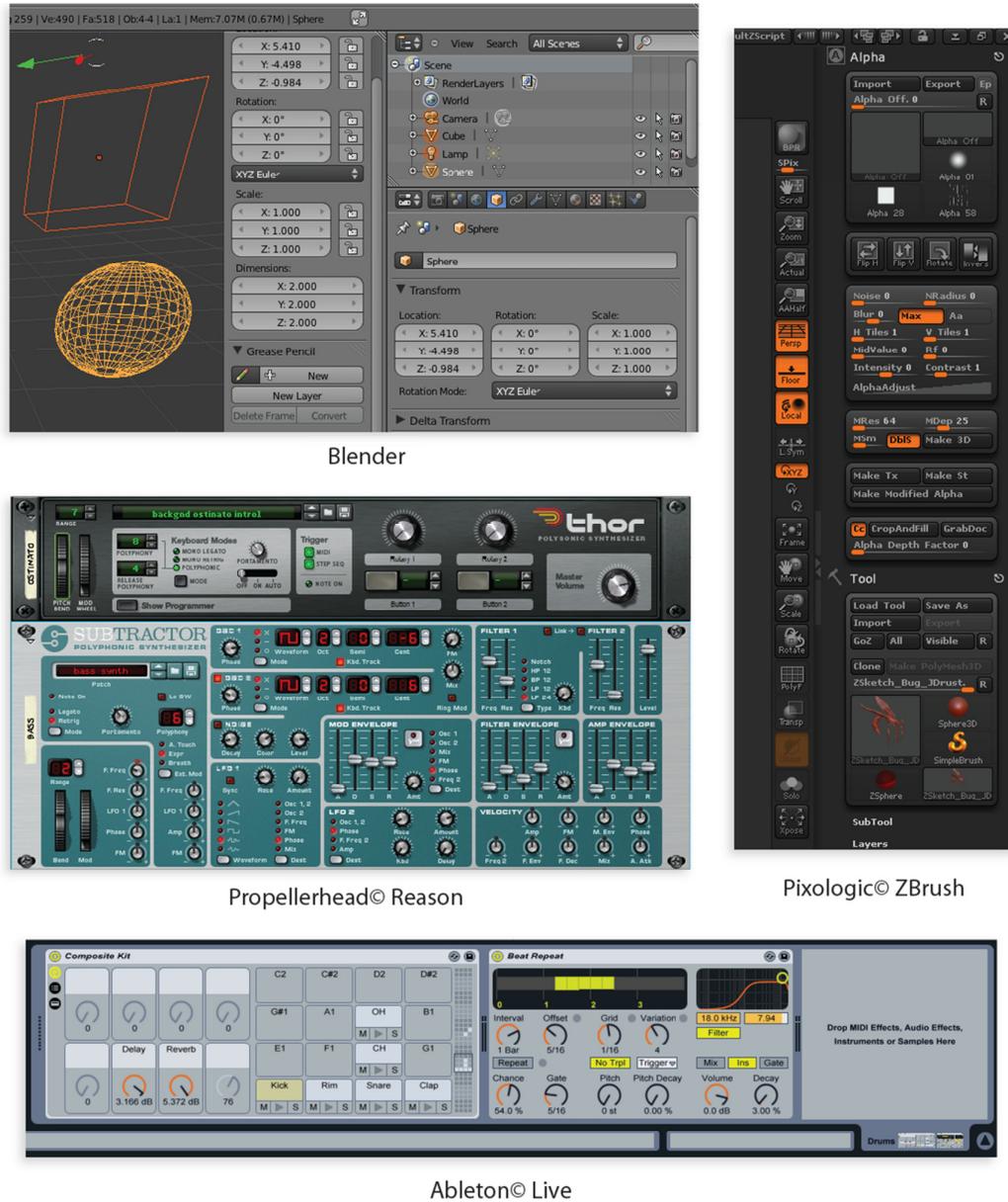
### ***2.5. Supporting Experts: dedicated techniques***

GUIs allowed human-computer interfaces to support needs of novices. But interfaces also needed to support the needs of experts who use applications repeatedly and want to “get things done”. As discussed in Section 2.3, experts need control and efficiency.

To provide a high degree of control, some applications chose to include a textual interface or scripting mechanism. In many cases, however, this became redundant because GUIs evolved over the years to a point where they could provide sophisticated control over computing processes with new types of GUI widgets, direct manipulation techniques, and strategies of organizing high numbers of GUI controls into meaningful conceptual and visual structures (see Figure 2.2). In fact, specialized interface controls often make it incredibly natural and easy to specify some parameters which would be otherwise difficult to define and manipulate.

To enable efficient interactions, a dominant approach is to include supplementary techniques that allow users to by-pass selections from GUIs with shortcuts, such as keyboard accelerators (also known as hotkeys), marks, gestures, and vocal commands.

The resulting situation is that current interfaces provide two separate modes, one graphical and one non-graphical for better performance. Note that the second mode provides access to a subset of the functionality provided by the first mode; the only goal is to make this access faster. It can be said that the two modes create a performance-related separation at the human-computer interface.



**Figure 2.2.** Four examples of current Graphical User Interfaces. Current GUIs can provide sophisticated control over application processing. Visual structures and specialized interfaces controls make it relatively easy to explore ‘what-if’ scenarios efficiently, regardless of the fact that the underlying functionality might be very complex.

This separation has a long history and is advocated by various HCI books and guidelines [Nielsen, 1992; Galitz, 2002]. For example, *Designing The User Interface*, by Ben Shneiderman and Catherine Plaisant, promotes menus and dialog boxes for

novices, while for experts “*String of commands, shortcuts through menus, abbreviations, and other accelerators are requirements*” [Shneiderman and Plaisant, 2004, p. 68]. This separation is also present in the current nomenclature where graphical menus are often considered and called *novice mode*, and shortcuts, *expert mode* [Bailly, 2007; Bau and Mackay, 2008; Roudaut et. al., 2009].

Shortcuts are necessary and prove efficient. However, the separation creates a situation where it is assumed that novices will use GUIs and experts will switch to shortcut techniques for better performance. And this assumption leads to the consequence that GUIs are not being designed with performance in mind. This observation is supported by many works which note the performance problems of traditional GUIs, such as toolbars [Kurtenbach et. al., 1999; Oel et. al. 2001] and menus [Kobayashi and Igarashi 2003; Cockburn and Gin, 2006; Tanvir et. al., 2008], which can be attributed to design properties rather than human capabilities.

This thesis focuses further on the above performance-related separation in the context of menus. Menus are not only the most prominent example of the above separation but are also one of the most widely used interaction techniques. We elaborate on both points below.

### ***2.5.1. Ubiquity and importance of menus***

In the context of the human-computer interface discussed in Section 2.2, menus are often a mode of choice in terms of flow of control [Norman, 1991b, p. 4].

In practice, menus appear in the majority of applications employing some sort of graphical user interface. The list of commands is a very generic and versatile way of representing and providing functionality. Other basic interaction modes, such as form fill-in, command languages, natural languages, questions and answers, and direct manipulation, are used to handle more specialized demands at the human-computer interface [Norman, 1991b]. It is also argued that the uptake of more innovative interaction styles, such as ubiquitous computing, pervasive computing, wearables, virtual reality, augmented reality, affective computing, and multi-touch, are to a large extent limited [Hartson 1998; Whittaker et. al., 2000; Beaudouin-Lafon, 2004]. As a consequence, applications employing other interaction styles often use them in conjunction with menus.

Commensurate with their importance, menus have received exceptional research interest. Lee and Raymond [1993] survey 112 menu-related papers written before 1993. Researchers have explored different menu structures [Snowberry et. al., 1983; Landauer and Nachbar, 1985], item ordering [Whalen and Mason, 1981; McDonald et. al. 1983; Dumais and Landauer, 1984], learning menu commands [Black and Moran, 1982; Card 1982; Kurtenbach and Buxton, 1994], marking menus [Kurtenbach, 1993; Zhao and Balakrishnan, 2004], cascading menu improvements [Kobayashi and Igarashi, 2003; Ahlström, 2005; Cockburn and Gin, 2006], adaptive menu techniques [Sears and Shneiderman, 1994; Findlater and McGrenere, 2004; Findlater et. al., 2009], visual search in menus [Byrne et. al., 1997; Hornof and Kieras, 1997; Aaltonen et. al., 1998], menu modelling [Byrne, 2001; Cockburn et. al., 2007; Ahlström et. al., 2010], and much more. Many user interface books have entire sections dedicated to menus [Galitz, 2002; Shneiderman and Plaisant, 2004]. There are even books fully devoted to the subject of menu selection [Norman, 1991b]. Major HCI conferences often include sessions fully devoted to menus.

### ***2.5.2. Menus as notable example of the separation***

Menus are probably the most notable example of the novice/expert separation discussed above.

As a novice mode, menus did not aim at providing experts with the best possible performance. It has been established that the most common cascading menus cause navigation problems [Kobayashi and Igarashi, 2003; Tanvir et. al., 2008], increase costs of errors [Samp and Decker, 2011], require precise steering interactions [Ahlström, 2005; Pastel, 2006], and generally result in slow performance [Cockburn et. al., 2007; Ahlström et. al., 2010].

Menus are typically supplemented with separate modes providing faster ways of invoking commands. Examples include keyboard accelerators, toolbars, marks, and gestures. Much research focuses on developing menu shortcut techniques that provide better performance [Kurtenbach, 1993; McLoone et. al., 2003; Zhao and Balakrishnan, 2004; Grossman et. al., 2007].

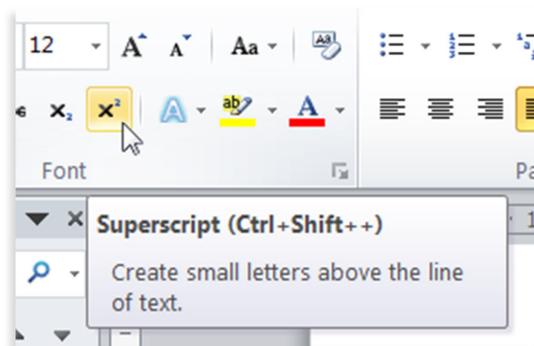
## ***2.6. Separation consequences***

As mentioned in the previous section, current software applications include both graphical menus for novices and shortcut techniques for experts. Undoubtedly, it is a

symptom of a good design that users with different level of expertise are supported. However, doing so by including two separate interface techniques causes some problems. In particular, shortcut techniques have some problems that are not immediately apparent but make them of limited utility. Below we review these problems.

### **2.6.1. Problem of self-revelation**

Shortcut techniques are not *self-revealing* [Dix, 1995; Kurtenbach, 1993; Grossman et. al., 2007]. Often times they are not visible – for example, gestures, marks, or to some extent hotkeys – and thus users might not know that they are available. At other times the intent of a shortcut technique might not be clear. For example, mnemonic keyboard assignments (i.e., hotkeys) displayed upon hovering over a menu item might not necessarily be meaningful to a novice user (see Figure 2.3). Or they might be simply ignored since once the menu is open, users already have an access to the items [Grossman et. al., 2007]. In the above cases, users will be condemned to the performance of graphical menus regardless of the amount of practice.



**Figure 2.3.** A toolbar tooltip shows a keyboard assignment of the superscript command (Ctrl+Shift++). This might not necessarily be meaningful to a novice user, or it might simply be ignored.

In a study employing CAD systems, Bhavnani et. al. [1996] observe that:

*“The overall attitude was to get the job done. The users had no way of knowing that there were better ways of executing a task. The help and document resources were voluminous, yet passive.”*

Learning efficient techniques might require explicit training or learning from peers [Bhavnani et. al., 1996; Tamborello et. al., 2006]. Peres et. al. [2004] show that shortcut usage can be influenced by social factors, such as working in an environment with other shortcut users. All in all, there exists extensive evidence showing that shortcuts, although available, are often not used [Peres et. al., 2004; Lane et. al., 2005]. Users often use a graphical interface alone to guide their actions [Gray, 2000; Fu and Gray, 2004].

### ***2.6.2. Problem of passive (vs incidental) learning***

It is established that novices rely on recognition while expert rely on recall (i.e., memory) to enable skilled performance. For example, unlike novices, experts remember locations of menu items or their hotkeys, and use this knowledge to perform fast selections that do not involve time-consuming visual search [Somberg, 1987; Kaptelinin, 1993]. Consequently, transitioning from novice to expert requires learning – e.g., memorizing item locations or hotkeys.

In the case of graphical menus, learning is incidental. Mere repetitions of the same selection result in memorizing item locations [Somberg, 1987; Kaptelinin, 1993]. This happens rather fast [Card 1982; Somberg, 1987; Kaptelinin, 1993; Cockburn et. al., 2007]. And each repetition improves retention [Card et. al., 1980; Calfee, 1975]. Shelton and Newhouse [1981] found that learning can occur during simple exposure to paired-associate stimuli even though participants were not requested to learn.

Learning shortcuts is typically harder because it is less incidental and more passive (or explicit) [Dix, 1995]. For example, hotkeys have to be memorized before their initial use. This type of learning process requires effort and attention [Calfee, 1975]. Human memory is relatively inefficient and slow when it comes to storing new knowledge in the long-term memory [Card et. al., 1983]. Consequently, users, who are aware of shortcuts, still need to invest a considerable amount of effort to learn them. This proves to be a barrier in the adoption of the shortcut techniques. Bhavnani et. al. [1996] observed that the users of CAD systems levelled-off in their learning and were using a system in suboptimal ways. Similar problems with learning shortcuts are also reported by [Peres et. al., 2004; Lane et. al., 2005; Grossmann et. al. 2007].

An interesting case is presented by marking menus. They are specifically designed to improve the learnability of their shortcut technique. A hand movement employed in selecting an item with a shortcut is the same as the movement when selecting an item

from a graphical menu [Kurtenbach, 1993]. Consequently, to some extent, the shortcut technique is learnt by using a graphical menu. Unfortunately, there are no studies comparing the learnability of the marking technique with other shortcut techniques.

Grossmann et. al. [2007] show that the learnability of hotkeys can be improved in two ways: 1) increasing the visibility and/or persistence of hotkeys; and 2) increasing the difficulty and/or the time cost of using a graphical menu to motivate use of hotkeys: for example, showing menu items as disabled or introducing explicit system delay. These approaches, however, increase the visual (or auditory) complexity of the interface and/or are based on degrading the usability of graphical menus.

Some work indicates that the lack of motivation to learn efficient methods can be explained with the cost/benefit analysis (CBA); that is, a relatively high cost/benefit ratio prevents the user from investing extra effort into learning shortcuts [Tamborello et. al., 2006]. It seems that the perception of benefits is especially important [Peres et. al., 2005].

### ***2.6.3. Problem of memory decay***

Because shortcut techniques rely on memory, they are not only difficult to learn but they are also subject to *memory decay* [Card et. al., 1983]. The meaning of the toolbar icons, hotkeys, marks, and gestures get forgotten if not used frequently. This is demonstrated empirically in [Kurtenbach and Buxton, 1994] where after lay off periods of few hours or days, experts needed to switch back to a graphical menu and re-learn what they forgot. Consequently, the shortcut techniques not only have to be learnt, they also must be maintained.

### ***2.6.4. Problem of perfect recall***

Graphical menus support partial or approximate recall. For example, if a user does not remember the exact location of a menu item but remembers that the item is somewhere at the top of the menu this approximate information can still improve search times. Similarly, approximate memory of other item features, such as the label length and command group, can help. By contrast, the shortcut techniques do not take advantage of the approximate memory. They typically require perfect recall; for example, an exact hotkey, mark, gesture or voice command. This makes the shortcut techniques more demanding to use.

### **2.6.5. Problem of sole reliance on memory**

There exists evidence that although experts rely on recall, they do not rely on it alone if they do not have to. Mayes et. al. [1988] found that:

*“[...] even experienced users can recall little of the menu contents, even though during use those menus are the instruments of their successful performance. It seems that the necessary information is picked up, used, and discarded; it is not learned in the sense that commands are learned. More exactly, users retain only enough information for recognition, not the much greater amount required for recall.”*

Payne [1991] conducted two menu studies and concluded that even experienced users must acquire the information they need from the display during interactions. Similar evidence is provided by O’Malley and Draper [1992].

The above studies demonstrate that even in routine actions experts do not rely on recall alone but also on recognition, if a graphical interface is provided. The optimal behaviour of experts is to distribute the load between the memory and the visual system. Shortcut techniques, however, often do not employ any graphical aids and thus force users to rely solely on recall. This increases memory load beyond the optimal (or at least the preferred) level and makes shortcuts potentially more demanding and difficult to use.

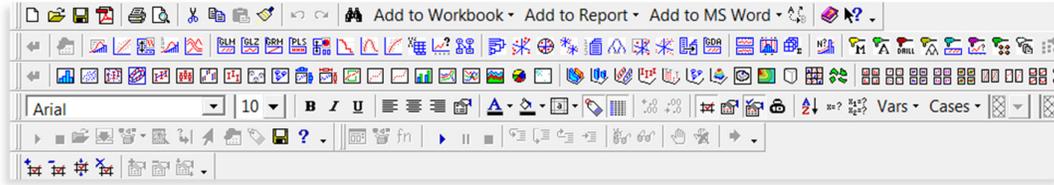
### **2.6.6. Problem of scalability**

Expert techniques often do not scale well. There are two aspects to this.

First, there is the memory limit. As the number of hotkeys, marks, gestures or icons increase, it becomes more difficult to remember all of them and to maintain this knowledge.

Second, there is the input space limit. The finite number of keyboard keys means that not all commands can have convenient and meaningful shortcuts, such as Ctrl-c for copy operation. Increasing the number of hotkeys will result in keyboard assignments that are difficult to remember and slow to articulate (e.g., Ctrl+Shift++ for the superscript command, see Figure 2.3). An increasing number of possible marks or gestures will make them physically similar or alternatively, will require making them more complex to ensure dissimilarity [Zhao et. al., 2006]. In both cases, however, either physically similar or physically more complex, the marks and gestures require more precision and are slower to articulate. As for toolbars, many commands

do not have clear iconic representations. Therefore, increasing the number of toolbar icons will result in awkward, meaningless icons (see Figure 2.4).



**Figure 2.4. Large toolbars consume much screen space and require the use of icons which do not clearly communicate associated commands.**

In practice, due to scalability issues, shortcuts are provided only for a limited number of commands. Therefore, not all commands can be executed efficiently. To mitigate this problem, applications often allow customization of which commands are assigned shortcuts. However, customization requires additional user effort. Empirical evidence shows that users are reluctant to customize [Mackay, 1991] even if this can bring about considerable performance improvements [Findlater and McGrenere, 2004].

### ***2.6.7. Problem of breaking structural organization***

Shortcut techniques break the structural organization of functionality. Issuing commands with hotkeys, marks, gestures or voice happens out of the context of command hierarchy or command groups. This prevents both the accidental learning of related functionality and learning the conceptual organization of functionality [Kurtenbach et. al., 1999]. Contrast this with graphical menus where selections are always made in the visual context of menu hierarchy and other related functions.

### ***2.6.8. Problem of availability***

Shortcut techniques often cannot be made available or their utility in some computing contexts might be degraded. Mobile devices, for example, may not provide toolbars due to their small screen size. Hotkeys are not an option for devices without a keyboard, such as some kiosks, terminal stations, and everyday appliances. Devices with small physical keys, a limited number of keys, or soft keyboards degrade the performance of hotkey execution. Single touch screens do not allow multi-key shortcuts. And devices without stylus interaction or touch capabilities do not enable marks and gestures.

The problems associated with shortcut techniques seem to be considerable. It becomes apparent that the importance of graphical menus is greater than could be initially asserted. Experts have to use graphical menus due to memory decay and scalability problems with shortcut techniques. And due to the problems of self-revelation, passive learning, sole reliance on memory, perfect recall, and limited availability, graphical menus are often all that is ever used.

The above problems lead to a situation in which graphical menus are used by experts to perform fast selections. It becomes apparent then, that the separation discussed in Section 2.5 leads to a wrong assumption that graphical menus are used only by novices and thus do not need to support fast selections. Also the nomenclature of calling graphical menus *the novice mode* is somewhat misleading.

What we advocate in this thesis is breaking the tradition of looking at graphical menus as only a novice technique. We postulate that graphical menus should provide better performance to support experienced users who do not transition, do not fully transition, or cannot transition to shortcut techniques. For this to happen, the novice and expert needs cannot be considered separately in the design of menus. Instead, the two need to be considered together.

It is important to note that such performance improvements do not seek to eliminate the need for shortcut techniques or to match their performance. They simply seek to improve the use in actual practice, where fast selections are attempted to be made from graphical menus.

## ***2.7. Potential benefits of faster menus***

Improving the performance of graphical menus could lead to a number of benefits.

The most apparent benefit is time savings – being able to do the job faster. Even small time differences can quickly add up to seconds and minutes since professionals spend hours each day using applications and invoking commands.

Another benefit is the decreased need for shortcut techniques. Many intermediate users fall somewhere between novice and expert on the expertise continuum (see Section 2.3) and, if graphical menus provided better performance, might never feel need for anything faster. The decreased need for shortcut techniques simplifies the

user interface and avoids the problems with shortcut techniques described in the previous section.

Yet another benefit comes from the fact that menu selections are not the user's main task. The menu selections can be considered a secondary task, a sub-task necessary to accomplish the main task. Making this sub-task seamless and faster will reduce disruption from the main task, thereby increasing productivity.

Probably the least apparent benefit is that improved performance can enhance creativity. The words of Mr. Brooks, explaining how computers improve self-expression, elucidate this point:

*“Most important, the new fluidity of the media [artistic drawings, building plans, mechanical drawings, musical compositions, photographs, video sequences, slide presentations, multimedia works, spreadsheets] makes easy the exploration of many radically different alternatives when a creative work is just taking form. Here is another case where an order of magnitude in a quantitative parameter, here change-time, makes a qualitative difference in how one goes about a task.*

*“Tools for drawing enable building designers to explore many more options per hour of creative investment. The connection of computers to synthesizers, with software for automatically generating or playing scores, makes it much easier to capture keyboard doodles. Digital photograph manipulation, as with Adobe Photoshop, enables minutes-long experiments that would take hours in a darkroom. Spreadsheets enable the easy exploration of dozens of “what if” alternative scenarios.”*

[Brooks, 1995, p. 280-281]

Performance improvements further increase this fluidity. They encourage broader exploration by lowering the cost of trying “what-if” alternatives.

The costs and problems of shortcut menu techniques described in Section 2.6 as well as the potential benefits of faster graphical menus gave rise to the question underlying development of this thesis. Namely,

*Can graphical menus be faster? Or have we rather reached the inherent performance limit that can be attained with the menu technique?*

## **2.8. Related work**

As mentioned in Section 2.5, slow performance of graphical menus is a recognized problem. This section presents related work focusing on improving the performance of graphical menus. This work can be divided into two categories. The first category focuses on ‘fixing’ the various problems of traditional cascading menus. The second category focuses on leveraging radial layout to decrease navigation requirements and improve selection performance.

### **2.8.1. Cascading menu improvements**

Cascading menus require dwelling over a parent item to open its sub-menu. This requirement slows down navigation. Kobayashi and Igarashi [2003] propose considering the direction of cursor movement in order to accelerate the opening of a sub-menu. Once the cursor starts moving towards the right direction, it is assumed that a user wants to access a sub-menu and it can be displayed straight away, without additional delay. The experiment, which employs selections from a four-level menu, demonstrates that the proposed menu reduced selection time by 12% and movement length by 31% when compared to a traditional cascading menu.

Cascading menus require precise horizontal steering movements between sub-menus. This type of pointing task is known to be more difficult and slower than typical Point-and-Click [Accot and Zhai, 1997; Ahlström, 2005]. Enlarged Activation Area Menus (EMUs) [Cockburn and Gin, 2006] and Adaptive Activation Area Menus (AAMUs) [Tanvir et. al., 2008], both cascading menu improvements, enlarge the steering area by using space from adjacent menu items, consequently decreasing the precision needed from a steering movement. For both menu types, controlled experiments demonstrated performance gains of approximately 22-23% compared to traditional cascading menus. The exact gain, however, will depend on the menu structure – i.e., the number and location of parent items, since this determines the extent to which the steering areas of parent items can be enlarged.

Ahlström [2005] propose using force fields to facilitate and accelerate steering movements. The force fields act within parent items, attracting the cursor toward a sub-menu. Ahlström presented an experiment employing selections from two- and three- level menus, and found that force-field menu decreased selection times by 18% compared to traditional menus.

Cockburn et. al., [2007] propose a *morphing menu* which dynamically adjusts the size of items according to their frequency of use. More frequently selected items become larger and thus easier and faster to select. Empirical data showed, however, that the performance of a morphing menu and a traditional cascading menu was not different.

A different approach to improving selections from a cascading menu is presented by *parallel menus* (also called *area menus* or *simultaneous menus*). These aim at decreasing navigation time by displaying more than one column of items simultaneously. Consequently, fewer sub-menus must be traversed to select a menu item. At the same time, however, visual clutter is increased along with distances to the menu items. Quinn and Cockburn [2008] presented an experiment which did not indicate any performance improvements of parallel menus over traditional cascading menus.

Aside from modifying menu appearance or behaviour, item selection performance can be also improved by modifying the menu structure or the order of menu items. Sears and Shneiderman [1994] show that a *split menu*, which places a small number of the most frequently selected items at the top of the menu, reduces selection times by 17% to 58% and yields significantly higher subjective preferences compared to a traditional cascading menu. It is important to note, however, that these results were obtained in an experiment which determined the frequency of item selections in advance. The item order never changed during the experiment. In practice, the frequency of item selections is often not known in advance.

*Spatially adaptive menus* adjust the order of menu items during use to ensure optimal performance based on the actual selection frequency. However, it turns out that such menus do not perform better, and often perform worse, than their static counterparts. The reason is that the continuous change of item ordering breaks the spatial consistency of menu layout, forcing users to re-search menu items rather than rely on remembered item locations [Somberg 1987; Mitchell and Shneiderman, 1989; Findlater and McGrenere, 2004, 2008; Cockburn et. al., 2007]. This problem is addressed by *graphical adaptive menus* which use visual cues rather than spatial cues to communicate predicted items. However, Gajos et. al. [2005, 2006] show that qualitative user feedback for such menus is negative; participants find dynamic changes to be disorienting. Furthermore, no performance results comparing graphical adaptive menus to static menus have been reported. Recently, Findlater et. al. [2009]

presented promising results for a new type of adaptation: not spatial, not graphical, but *temporal* where different onset times are used to indicate predicted items.

In any case, however, the effectiveness of an adaptive approaches greatly depends on how often users need to select the most frequently/recently selected items. Some studies demonstrate that menu selections tend to follow a *Zipf distribution* [Hansen et. al., 1984; Ellis and Hitchcock, 1986; Greenberg and Witten, 1993; Findlater and McGrenere 2004; Cockburn et. al., 2007]; in short, few items are selected much more often than the majority of the items. This hints that emphasizing the most frequently selected items (spatially, graphically, or temporarily) seems to be a good idea. However, in a Zipf distribution the long tail may account for as many selections as the head. Those selections will have to be made from a cascading menu. It is also possible that for different applications the frequency of item selections may vary.

Finally, adaptive and adaptable menus are predominantly concerned with improving the performance of the visual search, rather than of the navigation. As such, they aim at supporting novices more than experts. Furthermore, adaptive and adaptable menu techniques can be considered as more general techniques for improving the selection performance, not exclusive to cascading menus. The techniques used in the context of cascading menus are inherently exposed to their navigation problems.

### **2.8.2. Radial menus**

A considerably different approach is taken in the second category of work that aim to improve the performance of selections from graphical menus. Instead of ‘fixing’ cascading menus, the spatial organization of a menu layout is changed into a radial one. The rationale behind using the radial layout is that it optimizes the distances to menu items while also increasing item sizes. This enables faster selections. Although the idea of radial menu is very old and can be traced back to the work of Wiseman [1969], Newman and Sproull [1979] and Forsey [1984], development and research interest in this area is surprisingly sparse.

#### **2.8.2.1. Pie menus**

A pie menu is a single level radial menu. It involves placing the items along the circumference of a circle at equal distances from the center. Each item comprises a slice of pie. To perform a selection the user presses a button, moves the cursor only by

a small amount towards an indented item, and releases the button to finish the selection. In a controlled experiment employing a single-level menu with eight items, a pie menu yielded 15% faster selections and 42% fewer errors than a cascading menu.

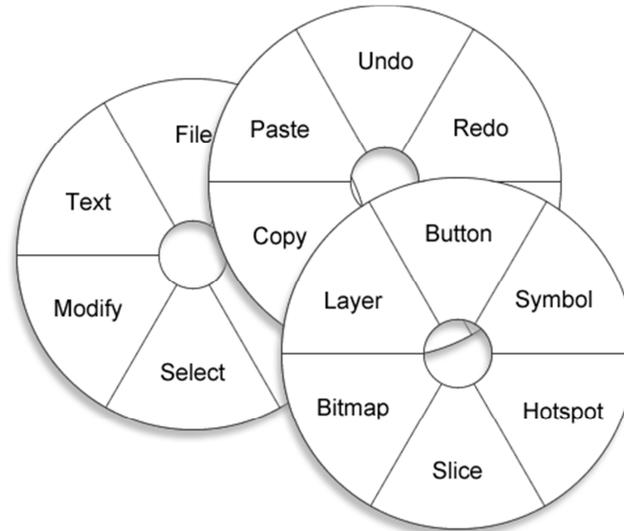
Pie menus support an additional selection technique called *mousing ahead*, which is intended for experienced users already familiar with the location of the menu items. Using this technique, a user performs a quick selection without looking at the menu. In this case, the pie menu is not displayed. The exact path of the cursor is not important. Mousing ahead relies only on the difference between the starting and the ending cursor position.

### **2.8.2.2. Marking menus**

After pie menus, work involving radial menus was done in the context of *marking menus*. Marking menus are similar to the mousing ahead technique. They add an ink-track to the cursor and use more sophisticated recognition in order to analyse the path of a cursor as a mark.

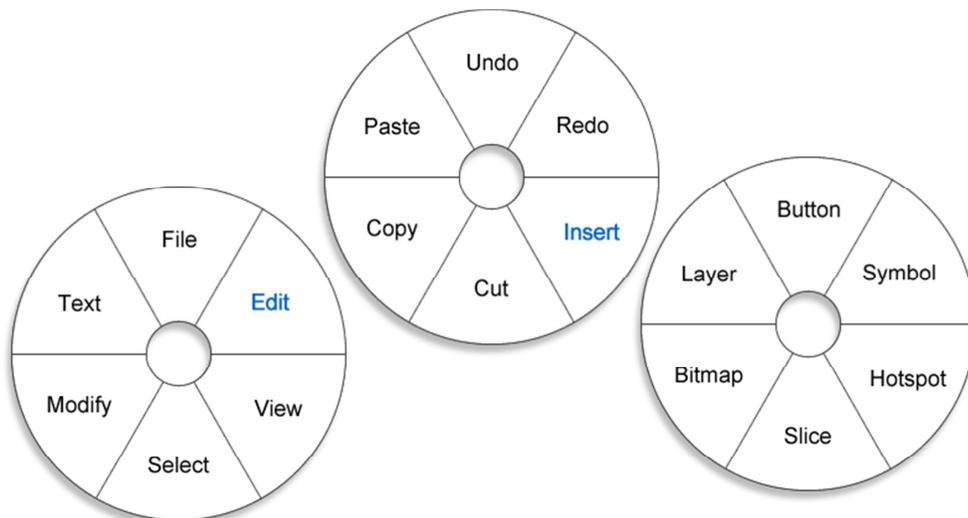
Selecting with marks, similarly to mousing ahead, is a shortcut technique that provides better selection performance for experts. Like any shortcut technique it is accompanied by a graphical menu. This graphical menu can be used by novices and by experts who have forgotten particular marks (i.e., shortcuts) to perform selections in a traditional way. A single-level marking menu uses a pie menu as a graphical menu. And hierarchical marking menus use a series of pie menus. Each menu level is represented by a separate pie menu and must be navigated separately. There are two strategies for laying out a series of pies:

- 1) Overlapping Pie Menus** (see Figure 2.5). In this case, the pies overlap. As a result the distances between the levels are shorter. However, overlapping pies obscure the content of the previous levels and make them more difficult to access. Consequently, OPM are problematic for novices who tend to traverse a menu hierarchy forth and back while searching for the right item. According to Bailly [2007] obscured view can lead to significant decreases in performance.



**Figure 2.5. Overlapping pie menus (OPM) decrease navigation distances between the levels but hamper access to the previous levels.**

**2) Non-overlapping Pie Menus** (see Figure 2.6). In this case, the pies are repositioned to avoid overlaps. This makes it better for novices since the previous levels are visible and accessible. However, the distances between the levels are increased thus impacting selection performance. Also screen consumption is increased.



**Figure 2.6. Non-overlapping pie menus (NPM) show the content of the previous levels but increase distances between the levels and screen consumption.**

It becomes apparent that OPM and NPM do not accelerate access to all menu items and levels. Additionally, it is only evident that a pie menu (either with or without overlaps) is faster than a cascading menu for small menus of eight items [Callahan et. al., 1988]. Increasing the number of items decreases each item's size, resulting in an increase in pie radius, which also effectively increases the distance to each item. Smaller and more distant items take longer to select [Fitts, 1954].

It is important to understand the reason for which pie menu and series of pie menus are chosen as a graphical mode for marking menus. The trajectories associated with selecting different items from a pie menu or a series of pie menus are considerably distinct (e.g., in terms of angle) [Kurtenbach, 1993]. Consequently, the same trajectories can be used as marks to uniquely identify different menu items. And because the trajectories are the same, a user can learn these marks by repetitive selections from a graphical menu. The reason for choosing pie menus as a graphical mode in marking menus is to improve the *learnability of marks*, and not to improve selection performance from a graphical menu.

Research on marking menus has focused predominantly on marks, their efficiency, accuracy, and scalability. The performance of graphical menus was not studied, nor compared to that of a cascading menu. The only exception is [Bailly, 2007] who compared novice performance with a number of marking menu designs.

The body of research focusing on improving marking menus is extensive. Various improvements often lead to different graphical radial menus or hybrid menus [Kurtenbach et. al., 1999]. However, similarly to OPM and NPM the goal of these graphical menus is to support learning marks and not to optimize performance of traditional selections from graphical menus.

### ***2.8.3. Summary of related work***

Research in the context of cascading menus focused on 'fixing' their problems, while research in the context of radial menus focused predominantly on marking techniques. The goal of this thesis is not to 'fix' problems of designs which were not built for fast selections. Rather, the goal is to design a graphical menu which among other requirements explicitly aims at improving the performance of traditional selections. Despite promising results for optimizing selection performance from single-level menus (i.e., pie menus) with radial layouts, there has not been any further interest on leveraging radial layout to improve selection performance from hierarchical graphical

menus. Our goal is to continue exploration in that direction, that is, how radial layouts can support graphical menu design, and in particular how they can bring about better selection performance.

The following chapters present a design of a graphical radial menu and a comprehensive evaluation of the proposed design as well as more general characteristics of radial layouts.



## ***3. Design***

This chapter presents a design of a graphical menu. We start with the definition of the design space. Then the design goals are presented. Next, we review menu characteristics which directly influence our goals. Finally, we describe design of the menu and explain our design decisions.

### ***3.1. Design space***

Design space is established by the basic definition of a graphical menu: a graphical menu provides two important elements: 1) graphical representations of application commands; and 2) a mechanism by which the users can choose one of these representations [Lee and Raymond, 1993] (see also Section 2.4). The cascading menus and radial menus reviewed in Section 2.8 all fall into this design space although their visual representation and selection mechanisms vary.

An example of a design that does not fit into this design space would be a voice menu. This type of menus is often used in phone services including mobile operator services and banking services. In a voice menu, all menu options are read aloud after which a user selects one of the options by pressing an appropriate button. In this case, a user relies on auditory rather than visual recognition and on memory for read aloud associations between commands and buttons.

### ***3.2. Design goals***

Our fundamental goals are to effectively support the needs of novices, experts, and those transitioning from novice to expert in a single graphical menu. The above goals are high-level. We now divide them into more tangible goals which can be more readily used to guide the design. These more detailed goals are defined in the context of novice behaviour, expert behaviour, transition from novice to expert, and other technical requirements.

### **3.2.1. Novices**

Current applications often contain hundreds and thousands of commands. And although these commands are organized into meaningful structures, finding a desired item is often not easy.

Novice behaviour with menus can be explained with *information foraging* theory which describes how people search for information [Pirolli and Card, 1999; Pirolli, 2007]. According to the theory, novices move to those parts of menu structure which have strong *information scent*. An assessment of the information scent of a menu item is based on its visual representation (e.g., label or icon) and with relation to the current task at hand. Users might not always make optimal choices; rather if the scent is arbitrarily satisfying a user might decide to follow it without checking other alternatives. It is not surprising that occasionally novices choose the wrong locations and do not find what they were looking for. In this case they revert back and proceed to another location of strong scent. Consequently, novices often navigate menus forth and back visiting a few sub-menus before finding an item. This is empirically demonstrated in [Norman and Chin, 1988] where novices visited 12 sub-menus on average while the optimal selection required traversing only four sub-menus. Snowberry et. al. [1983] provide supporting evidence reporting that 4 – 34% of selections in menu hierarchies are erroneous (i.e., wrong parent items selected). Tombaugh and McEwen [1982] observed that when searching for information, users were very likely to choose menu items that did not lead to the desired information

Novices are unfamiliar with the menu content and thus do not know the locations of menu items. They search for an item visually and then navigate to it [Cockburn et. al., 2007]. Typically, the visual search time dominates the total selection time [Samp and Decker, 2010]. When searching for a desired item, novices rely on readability and the meaning of the item labels and icons [Kaptelinin, 1993].

Lee and Raymond [1993] suggest that for novices, menus serve as documentation of an application's functionality. Graphical menus are designed to enable guided exploration and learning [Kurtenbach, 1993]. This mode of knowledge acquisition is not only preferred but also one of the most successful [Charney et. al. 1990; Franzke, 1995; Rieman 1996].

However, novices performing exploratory search in menus often experience a problem of *disorientation* [Robertson et. al., 1981; Mantei 1982; Elm and Woods,

1985; Yu and Roh, 2002; Alwis and Murphy, 2005] (also referred to as *navigation problem* [Kurtenbach, 1993, p.39] and *getting lost* [Dieberger, 1997; Yu and Roh, 2002]). This is because they not only have to follow an information scent but also have to remember which parts of a menu structure they have already visited. And if novices do not have a clear conception of relationships, do not know their present location, and find it difficult to decide where to look next, they feel disoriented [Robertson et. al., 1981; Elm and Woods, 1985]. Norman and Butler [1989] present a menu study demonstrating that participants searching for menu items tended to repeat paths and forgot where they had looked before.

To support novice behaviour the following design requirements have been defined (ordered by priority):

**Ease of use (EASY).** The goal is for the menu to be easy to use. Visual representations of the commands should be readable and clear to understand. The select mechanism should be simple, intuitive, and easy to learn. The menu should resemble the behaviour of a typical menu to the maximum possible extent: it should ‘feel’ familiar for novices who might have used other menu systems before and might have certain expectations.

**Guided exploration (GUIDE).** The goal is to help novices find and learn the functionality of an application through guided exploration. To this end, a menu needs to decrease the information load posed on a user confronting a large command space. The menu should lead the user toward a relevant command. And it also should support the behaviour when a novice makes mistakes and needs to explore multiple locations in the command space. The menu should prevent disorientation and let the user explore all relevant locations confidently and efficiently.

**Effective Visual Search (SEARCH).** The goal is to enable effective visual search. Novices should be able to quickly search menu content and newly visited locations.

**Effective Navigation (NAV).** The goal is to enable effective navigation. Novices should be able to quickly and seamlessly move between relevant locations and items.

### **3.2.2. Experts**

In contrast to novices, experts do not rely on the readability of menu items. In fact, experts have problems recalling labels of frequently used menu items [Mayes et. al., 1988]. Kaptelinin [1993] demonstrates that degrading readability through changing label characters into dots does not impact expert performance.

Experts do not search for menu items visually. They are familiar with menu content. They remember the locations of items and rely on these when performing menu selections. Somberg [1987] shows that for 6 blocks of 82 selections, positionally constant arrangement (i.e., consistent assignment of individual items to screen positions) is the most effective compared to three other arrangements: alphabetical, random, and probability of selection (i.e., the most popular choices are positioned near the beginning of the list). Similar results are demonstrated by [Kaptelinin, 1993; McDonald et. al. 1983; Seppälä and Salvendy, 1985]. Experts only decide which item to select and then navigate to it [Cockburn et. al., 2007]. Extensive menu use can lead to a substantial decrease in decision time [Card et. al., 1983; Lane et. al., 1993] and it is typically the pointing time that dominates the total selection time [Samp and Decker, 2010].

For experts, the menus serve as a way of accessing the functionality of an application. Expert behaviour can be best described as unprompted and efficient actions. Such actions are typically of less precision. The menu selection is merely a means of invoking the intended functionality.

To support the expert behaviour the following design requirement has been defined:

**Effective Navigation (NAV).** The goal is to enable effective navigation to menu items (note that this goal was also defined in the previous section). The menu should aim at shortening navigation times, require little navigation precision, and reduce navigation errors.

### **3.2.3. Novice-to-Expert transition**

The transition from novice to expert is associated with a transformation in the mental representation of the menu. While novices rely on readability, experts rely on global visual features such as the location of menu items; readability becomes unimportant. Cockburn et. al. [2007] demonstrates that the rate of novice-to-expert transition

depends on the stability of global features. If they are stable, the transition happens rather quickly [Kaptelinin, 1993; Seppälä and Salvendy, 1985]; notable differences occur even after one exposure to each menu item [McDonald et. al. 1983].

Experts rely on recall rather than recognition. Global features are memorized and only their successful retrieval enables users to avoid visual search. But due to memory decay, experts occasionally forget what they learnt. In such cases, they return back to novice behaviour; once again they have to search for items visually. This is demonstrated empirically in [Kurtenbach and Buxton, 1994] where after lay off periods of few hours or days, experts needed to switch back to novice techniques and re-learn what they forgot. Experts also return back to novice behaviour if memorized global visual features change: for example, when a memorized location of an item changes. This situation happens with spatially adaptive menus that constantly reorder menu items [Somberg 1987; Mitchel and Shneiderman, 1989; Findlater and McGrenere, 2004, 2008; Findlater et. al., 2009]. Cockburn et. al. [2007] show that the learnability of such menus is degraded. Finally, experts return back to novice behaviour when they want to find new functionality, previously not used or used sporadically.

In sum, the transition from novice to expert does not happen once and forever. Rather than being a novice or an expert, users often switch between both behaviours over time.

To support the novice-to-expert transition the following design requirement has been defined:

**Accelerate and sustain the novice-to-expert transition (N2E).** The goal is to accelerate rate of the novice-to-expert transition and to avoid forcing experts to return to novice behaviour. Although it is not possible to prevent memory decay, a design can avoid unnecessary changes of global visual features, such as item location, size, colour, and so on.

### **3.2.4. Miscellaneous**

Finally, some design requirements have been defined which do not directly address novice and expert behaviours, but rather represent some technical requirements:

**Quantity (QUANT).** Contemporary applications often contain hundreds of commands. Even context menus with a reduced number of commands can still contain tens and hundreds of these. The goal is to accommodate hundreds of commands.

**Reduced Screen Consumption (SPACE).** The goal is to limit the amount of screen space needed for the menu. This is especially important for devices with small screens such as smartphones.

### ***3.2.5. Summary***

The goals defined in this section will be used in the remaining of this chapter to explain our design decisions and ground them in previous research findings. It is important to note that design is a creative process and can be understood as a ‘best effort’ approach. This means that we make decisions which are the best for achieving our goals based on the previous findings and our subjective judgments. However, to support our central hypothesis (see Section 1.2) the goals NAV, SEARCH, and SPACE need to be investigated thoroughly in rigorous experiments. At the end of this chapter we define exact measures for the above goals and set the roadmap for the following experimental part.

### ***3.3. Design characteristics***

A menu has various characteristics. They can be viewed as design variables whose values are determined in the design process. An example of a characteristic is the size of a menu item. Some menu characteristics are particularly important for our design because their choice directly benefits or impacts the goals defined in Section 3.2. For example, a choice of the exact item size determines the effectiveness of navigation (goal NAV).

This section presents the characteristics most important for our goals. These characteristics are used to organize a review of the relevant research work which informed our design decisions. Each characteristic is discussed in terms of benefits and impacts on our goals. If a particular choice of a characteristic value benefits our goals, it is indicated in brackets using the following notion: goal X+, where X is a symbolic name of the particular goal being benefited. The symbolic names of our goals are provided in Section 3.2. Similarly, the notion: goal X- indicates that a particular

characteristic value impacts the goal X. For example, big menu sizes improve pointing performance (goal NAV+) but they also increase screen consumption (goal SPACE-).

There are nine characteristics. The first characteristic – menu structure – relates to the structural organization of menu items. The next four characteristics – navigation distances, discrimination method, spacing and overlaps, and item ordering and grouping – relate mostly to the spatial layout of the menu; that is, how elements of menu structure are represented and arranged spatially. The next three characteristics – item size and shape, visual communication, and navigation mechanisms and cues – relate to visual representation of layout elements. Finally, the last characteristic – interaction methods – relates to different interaction styles of selecting menu items.

### **3.3.1. Menu structure**

Applications often contain hundreds of commands. It is not an easy task to find relevant commands within so many alternatives. Robertson, McCracken, and Newell [1981] describe the performance of an early menu selection system named ZOG in a following manner:

*“Users readily get lost in using ZOG. The user does not know where he is, how to get where he wants to go, or what to do; he feels lost and may take excessively long to respond. This happens in all sorts of nets, especially complex nets or nets without regular structure”.*

To tackle these challenges, various approaches have been proposed to organize commands into different forms of menu structures including single menus, sequential menus, menu networks, and menu hierarchies [Norman, 1991b].

Over the years, one form surfaced as the most effective and currently also the most commonly used: a menu hierarchy [Norman, 1991b]. A menu hierarchy is a tree structure that contains two types of items: *structural items* and *functional items*. Structural items serve as categories which determine the organization of the menu structure. They are not associated with any application commands. They are parent items in a hierarchy and always lead to deeper levels that contain more specific items. Functional items represent specific application commands. They belong to categories but they never serve as categories which contain more items. They are leaves in a hierarchy, and they can be located on any of its levels.

A menu hierarchy facilitates exploration and visual search (goals GUIDE+, SEARCH+). Paap and Roske-Hofstrand [1986] observes that when the alternatives in a

menu are organized into categories, the scope of the search may be substantially reduced. This is because a user does not have to investigate the whole category before deciding if the desired item could possibly be there or not. In the context of information foraging (see Section 3.2.1), the scent of an entire category can be assessed based on its label or by scanning just a few items. If an item is unlikely to be there, a user can move to a different location. Use of menu hierarchy to guide the search is empirically demonstrated by Whalen and Mason [1981] who showed that miscategorised items are more difficult to find than items assigned to ambiguous categories and synonymous categories. Dumais and Landauer [1984] also provide evidence that the effectiveness of search highly depends on meaningful, unambiguous category labels. Snowberry et. al. [1983] show that menu hierarchy improves selection performance when compared to random ordering, up to a factor of two.

A menu hierarchy enables decreasing the amount of displayed information to only those hierarchy levels which are currently explored. This reduces screen consumption (goal SPACE+), decreases information load (goals EASY+, GUIDE+), and effectively supports exploratory search (goal GUIDE+) which we further discuss in Section 3.3.8.

Finally, because menu hierarchies are currently the most ubiquitous forms of menus, users are familiar with this structure and understand its semantics (goal EASY+).

### ***3.3.1.1. Finding the optimal menu hierarchy***

There has been a considerable amount of research focusing on depth versus breadth issue; that is, finding an optimal hierarchy structure. This research is predominantly based on two approaches: theoretical analyses and empirical studies (for a good review of the depth vs breadth issue see [Raymond 1986; Norman 1991b; Lee and Raymond 1993]). The conclusions are often inconsistent. However, one re-occurring empirical finding is that increasing the depth of the hierarchy has negative effects. Snowberry, Parkinson and Sission [1983] found that error rates increased from 4% to 34% as menu depth increased from one to six levels. Depth also impacts the selection performance [Landauer and Nachbar, 1985] and increases the costs of error recovery [Kiger, 1984].

At the same time increasing the depth has some positive effects. It can decrease the amount of screen space necessary to display relevant menu items (goal SPACE+), decrease information overload (goal EASY+), and help a user to “narrow down” the

choice and access items more quickly than using a flat list of items (goal GUIDE+) [Kurtenbach, 1993].

Some researchers have the opinion that efforts aiming at finding an optimal hierarchy incorrectly assume that such an optimum depends primarily on depth and breadth. Instead, [Lee and Raymond, 1993] have proposed that it is much more important for a menu hierarchy to reflect the structures inherently embedded in specific domains and the actions carried in the context of those domains. This view is supported by studies demonstrating that different categorizations and tasks – rather than depths and breadths – have a strong effect on user behaviour and overall performance [Whalen and Mason, 1981; McDonald et. al. 1983; Dumais and Landauer, 1984].

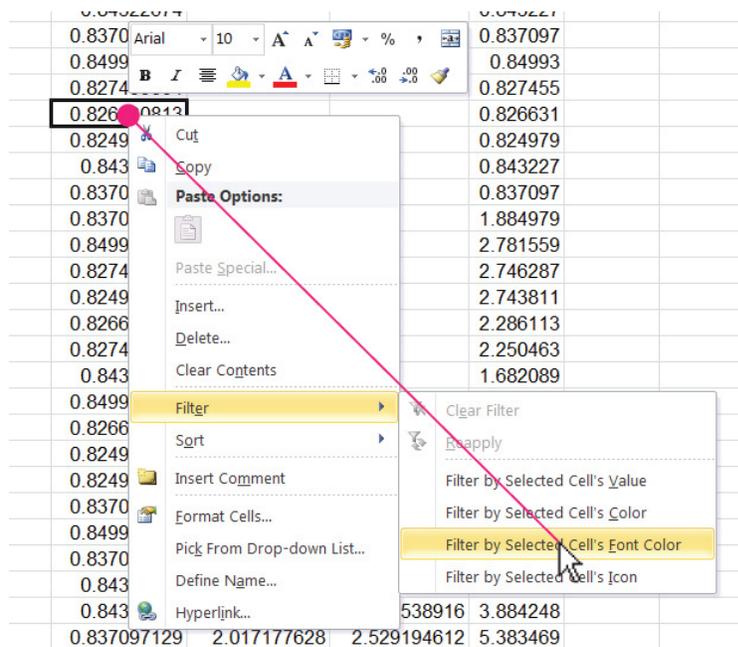
In practice, most applications have two- or three-level deep menus, with deeper levels having more menu items. Such menu structures are also the typical objects of current menu studies [Ahlström, 2005, Ahlström et. al., 2006; Ahlström et. al., 2010; Cockburn and Gin, 2006; Samp and Decker 2010; Zhao and Balakrishnan, 2004; Kurtenbach, 1993; Ellis et. al., 1995; Zhao et. al. 2006]. This lets us rephrase our goal Quantity (QUANT) in terms of menu hierarchies. Our goal is to accommodate at least three menu levels, with each subsequent level being able to accommodate more items.

### **3.3.2. Distances**

Graphical menus can use different spatial layouts which determine the distances between the menu structure elements: items and levels.

Shorter distances improve the navigation performance (goal NAV+) in four ways. First, shorter distances reduce the difficulty of a movement task, which reduces also movement time [Fitts, 1954]. Second, shorter distances decrease the pointer drift – that is, the distance between the initial and the final position of the mouse pointer on the screen after performing a selection from a menu (see Figure 3.1). The reduced drift means that the mouse pointer can be repositioned to its initial location faster. This is important in situations where the mouse pointer is used to directly manipulate content (e.g., many digital content creation applications). Third, short distances might allow for ballistic movements [Gan and Hoffman, 1988]. Such movements are motor programmed and during their execution visual feedback for path correction is not possible. Consequently, no visual control is involved and the movement can be performed faster. Ballistic movements occur if the duration of the movement is short,

around 200 ms [Gan and Hoffman, 1988]. They can also exceed 200 ms if the temporal-precision is precisely specified [Wright and Meyer, 1983]. Fourth, short distances require less positioning accuracy and thus result in fewer errors. Phillips and Triggs [2001] show that there is a significantly greater number of overshoot errors for far targets than for near targets. This can have a considerable impact on performance because, as Phillips and Triggs observe, participants overshoot their targets regularly. The authors also show that for far targets, a significantly greater proportion of the total movement duration is spent in deceleration than for near targets: this indicates that farther movements require more accuracy. Schmidt et. al. [1979] show that accuracy is better for short distances. Similarly, Thompson [2007, experiment 1] shows that accuracy decreases with greater distances, probably due to the increased movement velocity for farther targets.



**Figure 3.1. A selection from a two level menu. The pink circle and the pink line represent the initial position of the mouse pointer and the pointer drift respectively.**

Shorter distances, apart from improving navigation performance, can potentially also improve the performance of visual search (goal SEARCH+). To understand how this is possible, we first briefly discuss the visual search process: Visual search is best described as a process of successive *eye fixations* and *eye saccades* [Card 1982; Aaltonen et. al. 1998; Hornof and Kieras, 1997; Byrne 2001].

Eye fixation is when the eye remains still for some period of time – typically 120-600 ms – during which information is gathered and processed. The visual system is parallel in nature. When information is gathered, objects within the eye fovea are processed simultaneously. The fovea is a circular region within  $1^\circ$  of visual angle from the center of the gaze<sup>1</sup>. In the context of cascading menus, typically one to three items fit into the fovea, although this highly depends on the item sizes and the distance between the user and the screen [Hornof and Kieras, 1997]. The processing continues even after the gaze has shifted away. This ability allows people to quickly scan information while searching for a specific object.

Eye saccade is a movement connecting two fixation points. Saccades are fast, they last 30-120 ms and during that time practically no information is gathered. A sequence of fixations and saccades creates a scan path (see Figure 3.2).



**Figure 3.2. An example of a scan path consisting of saccades (lines) and fixations (circles). The example comes from an actual eye tracking study employing linear menus [Aaltonen et. al., 1998]. Courtesy of Antti Aaltonen and Aulikki Hyrskykari.**

Shorter distances can improve the visual search performance in two ways. First, shorter distances shorten the length of the scan path and thus fewer fixations and

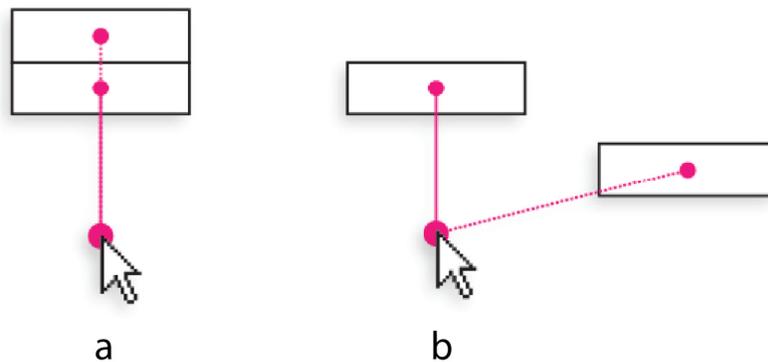
<sup>1</sup> For example, a square of 0.5 cm side length at a distance of 30 cm produces visual angle of  $1^\circ$ .

saccades are necessary. Second, shorter distances might potentially ‘squeeze’ more items into the eye fovea enabling more parallel processing. Hornof and Kieras [1997] show that more items in the fovea result in considerable improvements in search performance.

Finally, shorter distances keep the items and the levels tighter which decreases the amount of screen space necessary to display a menu (goal SPACE+).

### 3.3.3. Discrimination Method

Discrimination method determines a type of movement used to discriminate between menu items. Consider Figure 3.3.



**Figure 3.3. Example of discrimination by length and angle. In the first case (a), the items lay along the same direction and thus they are discriminated only by the movement length. In the second case (b), items are discriminated by the movement length but also by the movement angle.**

Callahan et. al. [1988] provides empirical evidence that for small eight-item menus, discrimination by angle enables faster selections than discrimination by length.

Different angles of approach affect selection performance [Whisenand and Emurian, 1999]. The fastest are the vertical movements while the slowest are the horizontal movements. However, although the differences are significant they tend to be small, approximately on the order of 50 ms.

The extent of the differences between items’ angles and positions determine how easy or difficult it is to properly discriminate between the items. For example, it requires more movement precision to discriminate between two items that lay at the

same angle and close to each other than between two items at two opposite angles. Kurtenbach and Buxton [1993] show that smaller angular differences lead to increased error rates and slower performance. The same is demonstrated by Zhao and Balakrishnan [2004]. Therefore, to improve performance (goal NAV), the design should preferably use discrimination by length and angle, and avoid small differences between items' angles and positions.

Discrimination by angle has a peculiar property: differences in angle can be determined more quickly than the differences in length. When discriminating by length, the total movement needs to terminate before it can be established which item the user intended to select. In contrast, when discriminating by angle, shortly after the movement commences, it can be often accurately predicted which item the user intends to select. This allows for very short navigation paths in pie menus. But also in other situations it might allow for predictions, which when coupled with appropriate techniques or visual cues, might enable faster selections, not necessarily terminating inside the desired item but well in advance (goal NAV+). In this context, possible techniques of interest include:

- a) *Sticky icons* [Worden et. al., 1997] and *Semantic pointing* [Blanch et. al., 2004] which dynamically adapt the control-display ratio, slowing down the cursor as it approaches a potential target.
- b) Techniques that expand the size of a target as a cursor approaches it [McGuffin and Balakrishnan, 2002; Zhai et. al., 2003].
- c) *Drag-and-pop* [Baudisch et. al., 2003] which reduces distance of movement by temporarily bringing potential targets closer to the cursor.
- d) Techniques such as *Object pointing* [Guiard et. al., 2004] and *Delphian desktop* [Asano et. al., 2005] which avoid the empty space between targets by making the mouse pointer jump from one object to another based on movement characteristics, such as direction or velocity peak.

The discrimination method also characterizes screen consumption because it determines how the menu expands spatially when it is navigated. Menus using discrimination by length expand in one direction. The cascading menu is an example; it expands rightwards and downwards. This allows the menu to be placed at the left and the top border of the screen. But also because of this, the menu requires significant space along the direction of expansion – i.e., on the right and the bottom sides – to

avoid repositioning or sub-menu overlaps. Such an expansion strategy might not be suitable for small screens.

Menus using discrimination by angle expand in more directions. A series of pie menus is an example; it can expand in all directions [Kurtenbach 1993; Zhao and Balakrishnan, 2004]. Such menus require less space on any side of the menu, thus are more suitable for small screens (goal SPACE+). At the same time, however, they might not be suitable for placement at screen borders.

### **3.3.4. Spacing and overlaps**

Spacing refers to empty space separating the elements of menu structure: items and levels. Spacing can be negative in which case elements of menu structure overlap.

Decreasing the spacing decreases the distances between menu elements, which can improve navigation performance (goal NAV+), visual search performance (goal SEARCH+), and decrease screen consumption (goal SPACE+) (see Section 3.3.2). This is an underlying strategy of the cascading menu improvement proposed by Kobayashi and Igarashi [2003]. It analyses the direction of the cursor movement. If the cursor moves towards a sub-menu of a parent item, that sub-menu is displayed immediately, next to the cursor, overlapping part of the menu. A study reported by Kobayashi and Igarashi [2003] employed four level deep menus and demonstrated that the proposed menu compared to a traditional cascading menu reduced selection time by 12% and movement length by 31%.

Overlaps can obscure items in the overlapped regions. Some items might not be visible, perhaps those that a novice should visit next. A special mechanism might be necessary to reveal the obscured region or to revert to a previous step in the selection process. Such a mechanism is necessary in marking menus that use series of pie menus [Kurtenbach, 1993, p. 56-57]. It is not self-revealing and might hamper ease of use (goal EASY-). Overlaps might also increase visual complexity and hinder the perception of the menu structure. All these problems hamper guided exploration (goal GUIDE-).

Spacing also influences the consequences of under- and overshooting errors – that is, errors resulting from selections that miss a target item. If the spacing is low – an example being a cascading menu where item borders are tangent – under- and overshooting errors may result in selections of items adjacent to the target item. The

consequence will vary on the activated command, but in some cases this might involve extensive processing or showing a pop-up dialog. The consequences of committing errors (or complexity of error recovery) have a considerable influence on user satisfaction [Feng and Sears, 2010]. Introducing some amount of empty space between the items diminishes the possibility of severe consequences of under- and overshooting errors (goal NAV+). This feature is important for experts who perform fast, unattended movements which are less precise and thus are more prone to missing a target item [Schmidt et. al., 1979; Wright and Meyer 1983; Zelaznik et. al., 1988; Zhou et. al., 2009].

In the same screen space, overlapping methods can display more menu content than non-overlapping methods. Therefore, overlapping methods might be considered if the screen space is limited. It might be more beneficial to use overlaps than to introduce another mechanism for dealing with content that does not fit in the screen (e.g., a scrolling mechanism).

### ***3.3.5. Item ordering and grouping***

Item ordering determines which command is assigned to which menu item. Research on item ordering has focused predominantly on two approaches: 1) analysing visual search strategies, and 2) empirical investigations of various ordering techniques.

#### ***3.3.5.1. Visual search strategies***

Visual search is often found to have elements of regularity. This means that during the search some of the items are systematically encountered before the others. It is sensible then to place the most important items in those locations which are searched first (goals GUIDE+, SEARCH+). It is also sensible to place the most important items in those locations which can be navigated the fastest (goal NAV+).

For example, in a cascading menu which organizes items in a vertical list, the visual search, although includes a random component, is predominantly top to bottom [Hornof and Kieras 1997; Aaltonen et. al., 1998; Byrne et. al., 1999]. The differences in search time between the first and the last item can be considerable. Samp and Decker [2011] report differences of 300–1500 ms depending on the menu size. Also navigation to the top items is faster by approximately 160-300 ms [Samp and Decker, 2011]. Sears and Shneiderman [1994] show that if the most frequently selected items are known in advance, placing four of them in the top of the menu reduces selection times from 17%

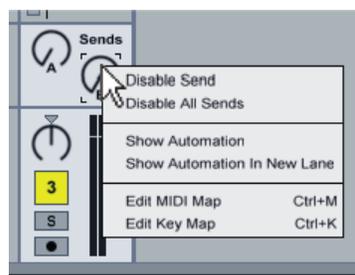
to 58%. If the frequency of selections is not known in advance, it might seem beneficial to adapt item ordering dynamically during menu use. However, as discussed in Section 2.8.1 and Section 3.2.3, such approach breaks positional constancy of the menu layout and forces users to re-search menu items (goal N2E-); that is, the users are no longer able to rely on remembered item locations. The approach of dynamic re-ordering does not improve performance, and often hinders it, compared to static counterparts [Somberg 1987; Mitchell and Shneiderman, 1989; Findlater and McGrenere, 2004, 2008; Cockburn et. al., 2007]. For this reason, to support novice-to-expert transition (goal N2E), dynamic changes to the ordering should be avoided or should be rare and subtle and lead to considerable benefits.

Findlater et. al. [2009] present a number of guidelines regarding designing adaptive menus (we add one point to this list):

- 1) Adaptive section should be small; Sears and Shneiderman [1994] propose up to four items.
- 2) Increase spatial consistency by copying rather than moving items from the static section to the adaptive section [Gajos et. al., 2005].
- 3) Decrease the frequency of adaptive changes [Cockburn et. al., 2007].
- 4) Adapt only if high accuracy can be assured [Gajos et. al., 2006, 2008; Findlater and McGrenere 2008].
- 5) Assure the predictability of the adaptive algorithm [Gajos et. al., 2008].

### **3.3.5.2. Empirical investigations**

Empirical investigations of item ordering and grouping have focused predominantly on three types of orderings: random, alphabetical, and functional (or categorical). The functional ordering displays groups of related functions (see Figure 3.4). In a sense, such a group could be perceived as an additional level of a menu hierarchy (see Section 3.3.1). However, in practice, groups differ from hierarchy levels in that they do not have a name and they use different presentation techniques.



**Figure 3.4. Ableton© Live. An example of functional ordering. Related functions are placed in a group. The groups are separated with horizontal lines.**

Random ordering is slower than alphabetical and functional [Card 1982; Liebelt et. al., 1982; McDonald et. al., 1983; Schultz and Curran, 1986; Hollands and Merikle, 1987]. Conclusions regarding alphabetical vs. functional ordering are inconsistent. One problem with alphabetical order is that it provides good performance results in experiments in which participants are presented with an exact name of a command to find. However, in real scenarios, novices often do not know a command name in advance. In experiments where a definition rather than exact command name is given, functional ordering is superior to the alphabetized one [McDonald et. al., 1983; Hollands and Merikle 1987].

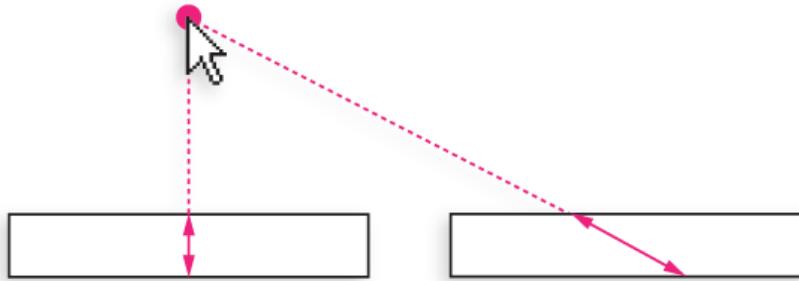
Functional ordering can facilitate development of a mental model of the menu and make learning the commands faster (goal EASY+, N2E+). Card [1982] presents empirical evidence that users organize menu items into memory chunks that reflect command groupings. In this sense, a menu group can be perceived as an additional feature by which experts can organize their knowledge and discriminate between items.

Functional ordering can also improve guidance and visual search performance (goal GUIDE+, SEARCH+). Item groups partition command space into smaller chunks of information. And since groups contain related functionality a user can assess the information scent of an entire group by scanning just a few items. Triesman [1982] found evidence that groups, rather than individual items, are scanned serially. Supporting evidence is also provided by McDonald et. al. [1983] who found that items grouped by category are searched faster than randomly ordered menus.

The effects of item ordering and grouping disappear with practice [Card 1982; McDonald et. al. 1983; Somberg, 1987]. Therefore they concerns only novices.

### 3.3.6. Item Size and shape

Size determines how big an item is. Many ways of measuring size of a screen object have been proposed, including object *height*, *width*, and *width×height* [MacKenzie, 1992a]. However, there exists evidence that calculating size of an object along direction of approach leads to the most accurate modelling results [MacKenzie, 1992a]. As an example, consider Figure 3.5. A long rectangle has arguably large size but if approached vertically from the top or the bottom the size of the box will match its height which turns out to be small.

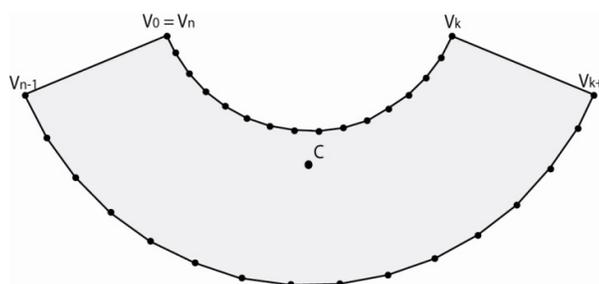


**Figure 3.5. The same rectangular item approached from two different angles. In the first case (the left item), the item is approached from the top, thus the item size equals the item height. In the second case (the right item), the item is approached from an angle, diagonally, thus the item size is larger.**

Maloney et. al. [2007] shows that mouse clicks can be biased towards or away from the center of the target based on the penalty associated with clicking on an adjacent target. This can decrease the effective size of menu items leading to lower performance. Additional spacing, which decreases penalty of under- and overshooting errors (see Section 3.3.4), might reduce these biases (goal NAV+).

Non-convex menu shapes might cause the perceived item center to be close to an item edge [Grossman et. al., 2007] (see Figure 3.6). In such a case under- and overshooting errors are more likely (goal NAV-). Moreover, non-convex menu shapes result in different item sizes depending on the direction of approach. For this reason, to support efficient navigation (goal NAV), convex, uniform shapes, such as circles and squares, are preferred. Their center is equally distant from all edges and their size is the same regardless of the direction of approach. Performance of circular and square items is either not different [Sheikh and Hoffmann, 1994] or the differences are found

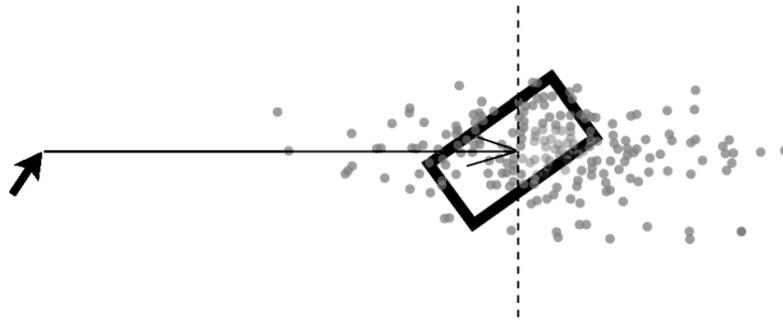
to be small, on the order of 20-50 ms, with smaller differences for bigger items [Whisenand and Emurian, 1997].



**Figure 3.6.** The center of the shape (point C) is close to the edge which increases the probability of under- and overshooting errors. This is an original figure from Grossman et. al. [2007]. Courtesy of Tovi Grossman.

Bigger item sizes improve navigation performance in three ways (goal NAV+). First, bigger sizes reduce the difficulty of a movement task (i.e., they decrease the accuracy constraint of a movement task) which also reduces its movement time [Fitts, 1954]. One reason for the increased accuracy with smaller items is the need for movement verification (i.e., a check at the end of a movement if the cursor is over the item). Smaller items increase the duration of movement verification [Walker et. al., 1993; Adam and Paas, 1996]. Graham and MacKenzie [1996] show that homing in on smaller targets is more difficult, thus more time consuming. Phillips and Triggs [2001] show that big targets require fewer submovements than small targets. They also show that significantly more time is spent in deceleration for the smaller items, which again indicates increased accuracy.

Second, bigger sizes reduce error rates. Series of rapid, aimed movements do not terminate at the center of an item but rather are spread around its perceived center [MacKenzie 1992a; Grossman et. al., 2007] (see Figure 3.7). The spread is often referred to as *endpoint variability*. There are more selections close to the center than far away from it. Smits-Engelsman et. al. [2002] show that endpoint variability is not sensitive to target size. Therefore, for small items, terminating locations are more likely to fall outside the item if a user tries to make fast and less accurate selections. Thompson [2007] shows that increasing target size decreases total error. Wobbrock et. al. [2008] show that target size has a strong effect on error rate, greater than that of target distance.



**Figure 3.7. Endpoint variability. Not all the terminating positions end up within the borders of the target item. This is an original figure from Grossman et. al. [2007]. Courtesy of Tovi Grossman.**

Third, larger item sizes are likely to induce users to employ speed/accuracy trade-off that favour speed over accuracy. Movement strategies are chosen to maximize expected gain, given the costs and benefits explicitly implemented in the environment [Trommershäuser et. al., 2005, 2008]. Humans take into account their own intrinsic motor variability [Dean et. al., 2007]. Even small changes in artefact design (the environment) make participants employ different *microstrategies* [Augustyn and Rosenbaum 2005, Gray and Boehm-Davis 2000] or speed/accuracy trade-offs [Hornof 2001]. Oel et. al. [2001] demonstrate that small targets, such as radio buttons and toolbar icons, need above-average time in order to be hit. Higher accuracy demands and error rates caused by smaller items might induce users to overly cautious movements that gain accuracy by sacrificing speed.

Although increasing item size might seem in general to be a good idea, typically this will also increase the distances between items and levels. Therefore, item size and navigation distance are coupled in a trade-off which needs to be properly balanced. It is felt that typical cascading menus have overly small items, typically 19-22 pixels [Ahlström 2005, 2010; Cockburn et. al., 2007; Samp and Decker 2010, 2011 journal], which is approximately three times smaller than typical desktop icons.

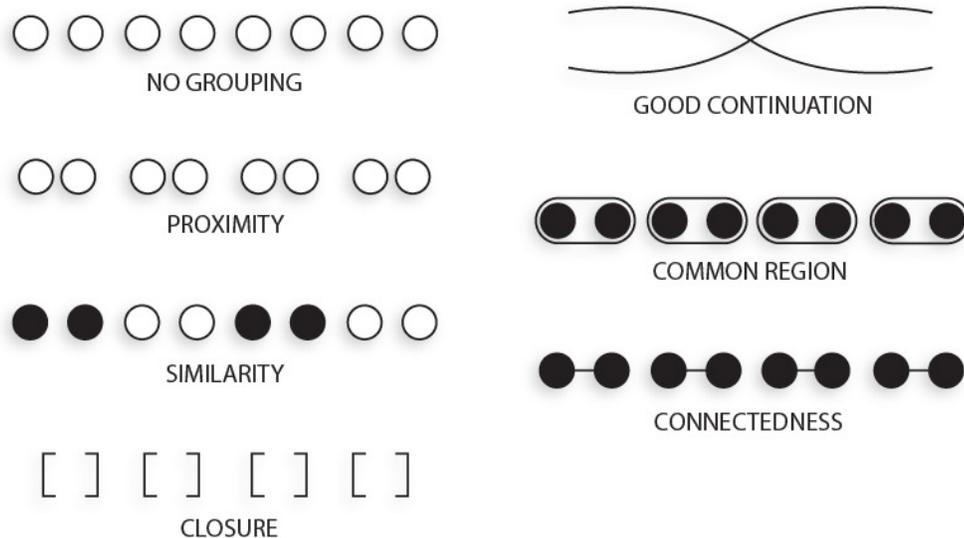
### **3.3.7. Visual communication**

Visual communication is concerned with creating visual representations of menu elements – items, item groups, and levels – that are effective and meaningful. Three aspects are communicated:

**Visual structure.** The visual representations communicate structure of the command space. This can help guide visual search (goal EASY+, GUIDE+, SEARCH+). Hornof [2001] provides empirical evidence that a clearly presented visual hierarchy, with distinguishable category items and their child items, guides visual search and results in better search performance compared to a visually unstructured list of items. The performance gain is considerable, 610 ms and 1350 ms for four and six groups of five items respectively.

The visual structure of a menu needs to effectively communicate concept of hierarchy levels and item groups. In cascading menus, for example, levels are communicated with disjoint lists of items and item groups are communicated with horizontal bars that separate two adjacent groups.

There is a whole body of research, conducted in the context of Gestalt psychology, devoted to the subject of human perception of item grouping [Wertheimer, 1958; Pomerantz 1981; Rock and Palmer, 1990]. Factors that effectively communicate the concept of perceptual group include proximity, similarity, closure, good continuation, common region, and connectedness (see Figure 3.8). Coren and Girgus demonstrate that perceptual groups strongly affect spatial memory. Participants underestimate distances within a perceptual group relative to distances outside groups.



**Figure 3.8. Laws of grouping.** This figure is reproduced from the original figure in [Rock and Palmer, 1990]. Reproduced with permission. Copyright © 1990 Scientific American, Inc. All rights reserved. Courtesy of Steve Palmer and Scientific American.

**Affordance.** Visual representations communicate the affordances of menu items. The term “affordance” comes from the perceptual psychologist J. J. Gibson [1977, 1979], who developed an “ecological” alternative to cognitive approaches.

*“Affordances provide strong clues to the operations of things. Plates are for pushing. Knobs are for turning. Slots are for inserting things into. Balls are for throwing or bouncing. When affordances are taken advantage of, the user knows what to do just by looking: no picture, label, or instruction needed.”*

[Norman 1988, p.9]<sup>2</sup>

The concept of affordances was introduced to the HCI community by Norman [1988] and since then, it has proved important for design [Gaver 1991; Norman, 1999]. In the HCI context, affordances can be best explained by example. Consider Figure 3.9. We say that buttons afford pressing but not moving because they protrude from the underlying surface.



**Figure 3.9. A button that protrudes from the underlying surface; it affords pressing rather than dragging or pulling.**

Well-communicated affordances improve ease of use (goal EASY+). Although it is perhaps difficult to objectively measure how well a menu item affords pushing, it is often possible to make a subjective judgment of which design alternative is the most appropriate. For example, non-uniform shapes and item overlaps might not clearly communicate to the first time users that each item is a separate command that can be selected.

**Item semantics.** Visual representations communicate the content of structural items (i.e., parent items) and the functionality associated with functional items. The most

---

<sup>2</sup> The presentation of the concept of affordances is shortened here for brevity. In particular, we do not discuss “perceived” vs “real” affordances. There is also some discussion regarding the inconsistent use of the concept [see McGrenere 2000, Norman 2011].

common representations are short textual labels (typically two/three words long), icons or combination of the two. Meaningful representations are important for novices. They improve ease of use, guidance, and performance (goal EASY+, GUIDE+). For example, Byrne [1993] demonstrates that simple icons (those discriminable based on a few features) help users in a search process, while complex icons are no better than simple rectangles. Whalen and Mason [1981] show that meaningless category names lead to the most severe errors.

Construction of meaningful command representations is not the subject of this thesis. There is a whole body of research focusing on this topic [Black and Moran, 1982; Hemenway, 1982; Furnas et. al., 1987; Goonetilleke, 2001]. From the current perspective it is important for our design to be capable of accommodating typical command labels and iconic representations.

### **3.3.8. Navigation mechanisms and cues**

#### **3.3.8.1. Focus + Context techniques**

A family of *Focus+Context* techniques (and closely related *Overview+Detail* techniques) explicitly aim at supporting exploratory search in structured information spaces (goal GUIDE+) [Furnas, 1986; Lamping et. al., 1995].

*“[...] a strategy is to provide a balance of local detail and global context. Local detail [focus] is needed for the local interactions with a structure, whether that means finding the nearest mailbox in midtown or editing a particular line of a large program. The global context is needed to tell the user what other parts of the structure exist and where they are (e.g., the Rockies are out west, beyond Chicago but before LA; there is an if construct above the else construct currently being edited). Global information may also be important even in the mere interpretation of local detail (e.g., the meaning of the else statement in fact depends on the content of the associated, but remote, if() statement).”*

[Furnas, 1986]

Focus+Context techniques are often coupled with a navigation mechanism which allows users to jump to any part of the context directly rather than forcing users to pass back through all the previous exploration steps. Such mechanisms allow the users to find the information more quickly [Parkinson et. al., 1988] and minimizes the number of navigation actions [Tauscher and Greenberg, 1997]. This contributes to goals EASY and NAV.

Focus+Context techniques often include also *navigation aids* [Yu and Roh, 2002]. Examples include maps or overviews [Conklin, 1987; Nielsen 1990], fisheye views [Furnas, 1986; Bederson 2000], contextual navigation aids [Park and Kim, 2000], and local navigation aids [Park and Kim, 2000]. These techniques directly address the problem of disorientation (see Section 3.2.1). They help users perceive their present location and help them decide where to go next. The aids improve search performance and learning (goal EASY+, GUIDE+, SEARCH+) [Nielsen, 1990; Park and Kim, 2000; Yu and Roh, 2002]. The subject of navigational and structural cues has been extensively researched in the context of hypertext and web systems because they involve exploring complex information structures. However, navigational problems also concern menu structures [Yu and Roh, 2002].

### **3.3.8.2. Unfolding menus**

Hierarchical menus support the concept of Focus+Context through the notion of *unfolding menus*—a cascading menu being an example. In an unfolding menu, a user is initially presented only with the first level of the menu hierarchy and is given a mechanism to “unfold” subsequent levels. This way, a user traverses menu in a “walking manner”, selecting parent items, and moving to their newly displayed sub-menus.

In an unfolding menu, the *focus* is the currently visited menu level while the *context* is constituted by all the previous levels and their items. Note that some techniques employ dedicated visual representations of the focus and the context, or use distortion techniques to emphasize them; for example fisheye menus [Bederson, 2000].

The unfolding approach to navigating hierarchical menus has a number of advantages. It decreases information load, visual complexity, and screen consumption by showing only selected hierarchy levels (goal EASY+, SPACE+). It also guides a novice (goal GUIDE+) by structuring the process of selecting an item into steps, each associated with a small portion of new information used to make a simple decision of where to go next. Quinn and Cockburn [2008] demonstrate that a fully parallel menu displaying all menu items at once, although it decreases navigation demands, is not better than a traditional unfolding menu. This hints that unfolding menus indeed effectively guide users in the selection process.

Hierarchical menus typically use three navigational aids. The first one is *selection preview*. It provides feedback indicating which item is about to be selected. The most common approach is to highlight an item directly under the mouse cursor. The second navigational aid is *path highlighting*. It marks all the parent items that lead to the currently visible level. Occasionally, some hierarchical menus also use a third navigational aid, which is the *level preview*. Once a cursor is over a parent item a preview of the corresponding sub-menu is displayed immediately without waiting for the selection. The preview of a sub-menu is typically dimmed. Visual preview of what is coming can improve response time, first demonstrated by Dodge [1907], further explored in the context of reading [Rayner, 1978; Fine et. al., 2001] but is also beneficial in HCI where it improves search performance and error rates [Kopetzky and Mühlhäuser, 1999; Woodruff et. al., 2001; Kristensson and Zhai, 2007]. These three navigational aids help users perceive their present location and help them decide where to go next (goal GUIDE+). They function as *cognitive landmarks* denoting focal points at which fundamental turns have been taken [Norman, 1991b]:

*“When we give directions to geographical locations, we generally use landmarks. If one gets lost, one can return to the landmark in order to redirect search. Major intersections, highways, and shopping centers serve this purpose. Similarly, in menu selection systems, users who have a sufficient mental model of the system use its cognitive landmarks to find their way around. When they get lost or fail to find a target item, they tend to return to focal point of the system”*

[Norman, 1991b, p. 230]

### **3.3.9. Interaction methods**

Interaction methods determine the type of motor actions used to select menu items.

According to the flow of information at the user interface described in Section 2.2, selecting a menu item looks as follow: a user first finds a command visually, then decodes the meaning of the visual representation, decides if the command should be activated, determines what information needs to be provided to activate that particular command, and finally inputs that information through the user interface. The efficiency of an interaction method depends on two factors: 1) how fast a corresponding input information can be determined for a given command, and 2) how long it takes to input this information.

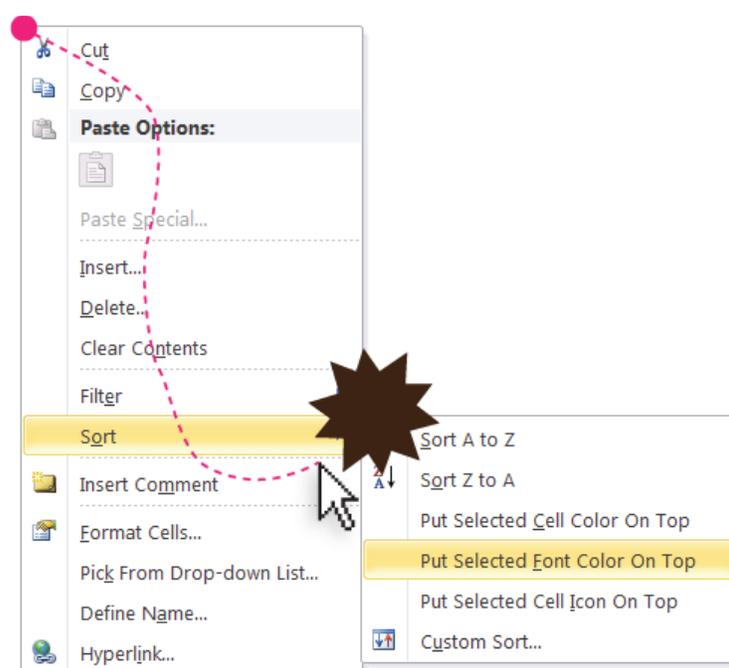
The most commonly used mechanism for issuing selections in graphical menus is Point-and-Select: a user points at a desired item and selects it, for example, with a button press. In this case, determining how to select a found item is straightforward; it does not require any conscious action. Compare this with a selection mechanism where a keyboard key has to be pressed to activate a command, and that key is the one corresponding to a character underlined in a command name. In this case, more cognitive processes are involved into determining what information needs to be input. Point-and-Select can be viewed as a technique with high compatibility between the stimulus (i.e., visual representation of an item) and the response (i.e., pointing at that item). Wallace [1971] would say that Point-and-Select requires little coding and recoding in order to map a stimulus to a response (S-R). The topic of S-R compatibility has received considerable amount of interest, predominantly in experimental psychology. Numerous studies demonstrate that high S-R compatibility significantly decreases response times and error rates [Fitts and Seeger, 1953; Chua et. al., 2003]. In this light, Point-and-Select is beneficial for the goals EASY and NAV.

Menu selection through Point-and-Select requires minimal effort regardless of the visual and functional complexity of a command being selected. Lee and Raymond [1993] refer to this advantageous characteristic of menus as *decoupling functions of input and display*.

**Point-and-Click vs Drag-and-Drop.** Menu selection through Point-and-Select can employ two different interaction styles. The first style is Point-and-Click, found, for example, in traditional single-level linear menus. A user first moves a pointer to a target item and then presses a button to select it. The second style is the Drag-and-Drop interaction style employed, for example, by marking menus [Kurtenbach 1993 thesis]. In this case, the user first presses a button, then moves to a target item (with the button still pressed), and finally releases the button to select the target. The Point-and-Click interaction style is faster, less error prone (goal NAV+), and preferred over the Drag-and-Drop interaction style (goal EASY+) [MacKenzie et. al., 1991]. Inkpen [2001] demonstrates similar findings in a study conducted with children.

**Dwell vs Button press.** There are also two different ways of issuing selections: 1) dwelling – e.g., used in cascading menus to select parent items; and 2) clicks – e.g., used in cascading menus to select a command. Mouse clicks are faster than typical dwell times (goal NAV+) (see Chapter 4). More importantly, however, dwelling as a way of selecting parent item restricts the trajectory and characteristics of subsequent

mouse movements. Consider Figure 3.10. A user attempts to navigate to the item “Put Selected Font Color On Top” in the sub-menu of the parent item “Sort”. But the trajectory of a mouse cursor crosses over the item “Insert Comment”. If the cursor stays long enough over the item “Insert Comment”, the system will interpret this as an intention to select that item. Consequently, the previous sub-menu will disappear and the user will not be able to select the desired item. This kind of error is not uncommon [Kobayashi and Igarashi, 2003; Cockburn and Gin, 2006; Tanvir, et. al., 2008]. And to avoid it, the trajectory of a mouse cursor should run inside the parent item. Such movements are more difficult and slower than non-restricted movements [Accot and Zhai, 1997].



**Figure 3.10.** A user tries to navigate to a sub-menu of the parent item “Sort”. However, because the cursor does not stay exactly inside the parent item, the sub-menu disappears.

Another technique of issuing selections is that of border crossing [Kurtenbach, 1993]. A user moves between items and selects one by moving the cursor outside the menu through the border of the target item. This approach is less intuitive (goal EASY-) and also restricts the movements trajectories (goal NAV-).

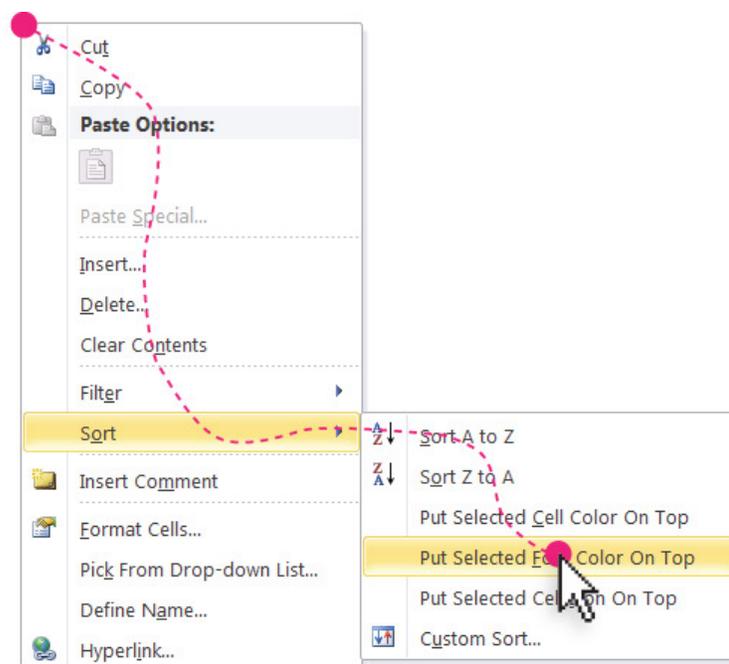
### **3.4. Design description**

This section describes the menu design. The presentation is divided into sections, each describing a single element of the design. We explain our decisions, referring back to the discussions on relevant characteristics presented in Section 3.3 and goals presented in Section 3.2.

There are six design elements. The first two – level arrangement and item arrangement – relate to the spatial arrangement of levels and items. The next three elements – item representation, level representation, and item grouping – relate to the visual representation of items, levels, and item groups. The last element – menu navigation – relates to interaction mechanisms and control.

#### **3.4.1. Levels arrangement**

Figure 3.11 presents a linear layout used in cascading menus. We can now analyse it in terms of the spatial menu characteristics presented in Section 3.3: navigation distances, discrimination method, and spacing and overlaps. The most common cascading menus require a sequence of vertical and horizontal movements [Ahlström, 2005], which is not the shortest path between items. Some designs reduce the distances by allowing diagonal movements [Cockburn and Gin, 2006; Tanvir et. al., 2008] or by overlapping levels [Kobayashi and Igarashi, 2003]. In any case, however, these reduced distances considerably exceed the distance to the closest free space around the starting location (goal NAV-, see Section 3.3.2). For example, the left and the top sides of the item “Cut” and the left side of the parent item “Sort” (see Figure 3.11) are not occupied despite being close. It would appear then that there is room for improvement regarding these distances.

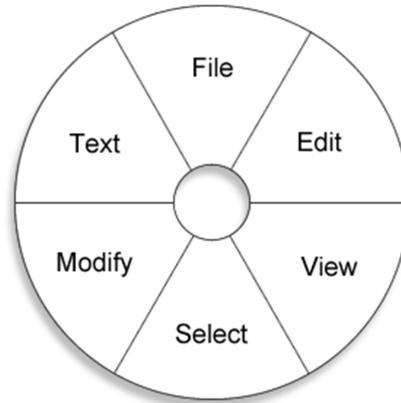


**Figure 3.11. A cascading menu using linear layout. An example of a selection consists of a sequence of alternating vertical and horizontal movements.**

Linear layout predominantly uses discrimination by length (goal NAV-, SPACE-, see Section 3.3.3). There is little variation in angle, even if the menu allows diagonal movements. And importantly, the differences in movement length are small because items are approached vertically and their height is typically in the range of 19-22 pixels [Ahlström 2005, 2010; Cockburn et. al., 2007; Samp and Decker 2010, 2011 journal]. Small items hamper performance and increase the probability of under- and overshooting errors, especially if fast and less precise selections are performed (goal NAV-, see Section 3.3.6). Cascading menus do not introduce any spacing between the items. Therefore, the consequences of under and overshooting errors can be high; that is, adjacent items can be accidentally selected (goal NAV-, see Section 3.3.4).

One layout that aims at reducing distances is the radial layout used in pie menu [Hopkins, 1987, 1991, Callahan et. al., 1988] (see Figure 3.12). The entire space around the center is utilized. The distance to each item is equal and short (goal NAV+, see Section 3.3.2). The items are discriminated by angle (goal NAV+, see Section 3.3.3). The radial layout has an additional advantage of increasing the item sizes, which we explore in the following sections. In a controlled experiment a pie menu turned to be

15% faster and 42% less error prone than a linear menu [Callahan et. al., 1988]. However, the pie menu is only a one level menu.



**Figure 3.12.** A pie menu using a radial layout. Each item is equidistant from the center.

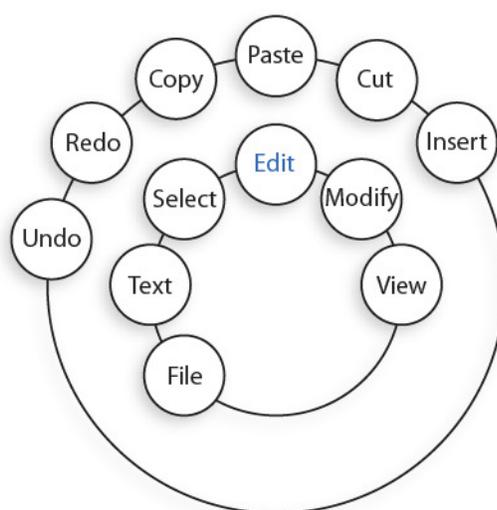
To support hierarchies, it has been proposed to use series of pie menus. However, this proposal was not motivated by the goal of improving selections from a graphical menu but rather to support marks (see Section 2.8.2 for more details). For example, series of pie menus either do not decrease the distance between the levels or they result in overlaps.

To build an efficient layout for hierarchies we looked into the fields of graph drawing and information visualization. One of the subjects in these fields is the creation of layouts adhering to various criteria, such as minimizing screen consumption and efficient communication of parent-child relationships. An approach called radial drawing [Eades 1992; Bernard 1994; Six and Tollis, 1999; Bernard and Mohammed, 2003; Jankun-Kelly and Ma, 2003; Giacomo et. al., 2004; Gansner and Coren, 2006], based on a concept of *concentric rings* (also referred to as *circles* or *orbits*) seemed to adhere to our criteria.

In radial drawing a central node is the focus, and the rest of the nodes are placed in concentric rings around the center according to some distance metric. The distance of a node from the focus determines the ring to which that node belongs. The nodes closest to the focus node are placed on the first ring, and farther nodes are placed on subsequent rings. Such visualizations are often coupled with a mechanism which allows users to interactively change the focus node. The new focus node is moved to

the center and the visualization is reorganized accordingly; the transition can be animated to improve user comprehension [Yee et. al., 2001].

To support hierarchical menus we use the concept of concentric rings in a different manner. The center is the menu root. The first ring corresponds to the first level of the menu hierarchy. However, the content of the subsequent rings does not depend on the distance from a focus node but rather on the user interactions. Each concentric ring represents a submenu of a selected parent item from a previous ring. And as will be explained in Section 3.4.6, the focus is fixed: selections do not change the focus but rather they determine what is displayed on the rings.

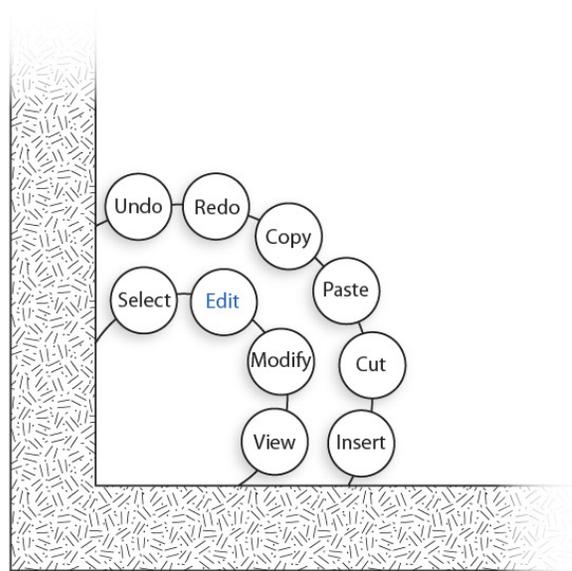


**Figure 3.13.** A radial layout based on a concept of concentric rings. Starting from the center, each subsequent ring represents a deeper menu level. The first ring contains items from the first menu level. Each subsequent ring contains the child items of the parent item selected on the previous level.

The concentric ring layout has a number of advantages related to the goals defined in Section 3.2. First, it supports hierarchies (goal EASY+, see Section 3.3.1). Second, it decreases the distances between the items and the levels (goals NAV+ and SEARCH+, see Section 3.3.2). Third, it combines discrimination by length and angle (goal NAV+, see Section 3.3.3). Fourth, the concentric ring layout prevents overlaps and preserves visibility of the levels and their items (goals EASY+, GUIDE+, SEARCH+, and NAV+, see Section 3.3.4 and Section 3.3.7). Fifth, the layout expands in all directions equally when the number of levels increases. Therefore, in some sense, the screen space is

preserved (goal SPACE+, see Section 3.3.3) when compared to layouts which lay deeper levels always on one side – e.g., the layout used by cascading menus. Sixth, the layout can accommodate an increasing number of items on subsequent levels because the radiuses of the corresponding rings increase (goal QUANT+).

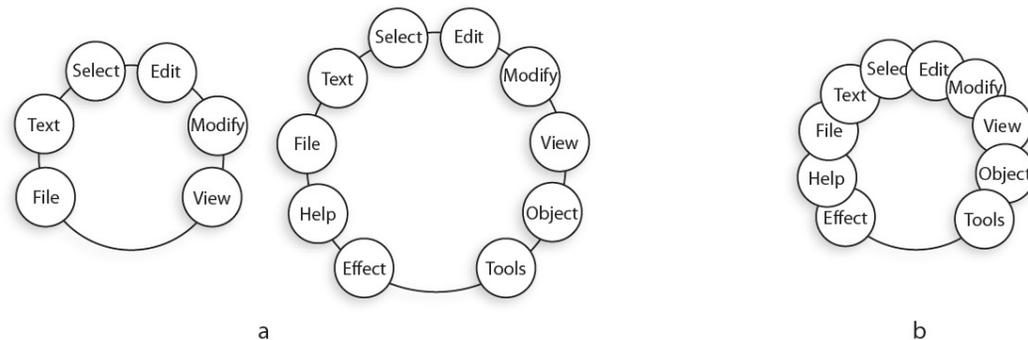
Because the concentric rings expand in all directions, it is undesirable to place them close to the screen borders (see Section 3.3.3). Doing so may partially obscure the menu or require repositioning. The layout is suitable for pop-up (or context) menus, typically invoked within the working space, away from the screen borders. It also makes accessing menu content faster by avoiding round trips to screen borders and reducing pointer drift (goal NAV+, see Section 3.3.2). Alternatively, if the menu structure allows, a concentric ring menu could sit close to the corner of the screen and expand as a slice of pizza toward the center when navigated (see Figure 3.14).



**Figure 3.14. A concentric ring menu sitting in the corner of the screen and expanding as a slice of pizza toward the center of the screen when navigated.**

The spatial capacity of the first ring might be too restrictive at times (goal QUANT-). Depending on the application a combination of the following remedies can be used: a) smaller item sizes; b) changing the structure of the menu, such as adding an intermediate first level with fewer items; c) moving the first ring away from the center to increase its spatial capacity; d) overlapping the items (see Figure 3.15). All four approaches are associated with some costs. Making items smaller make them more

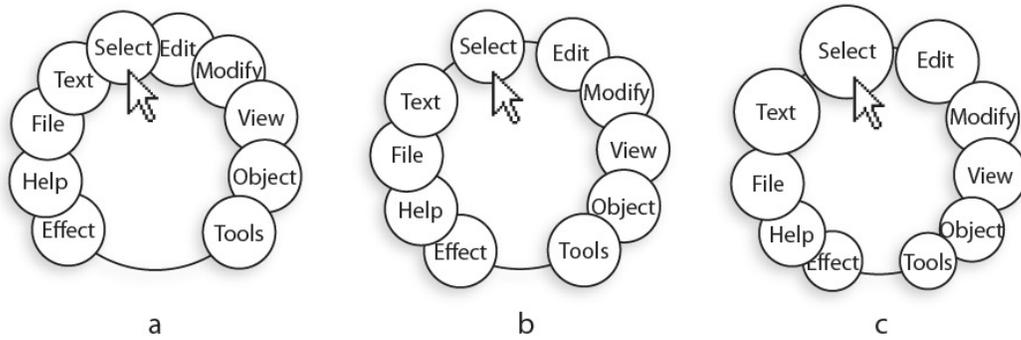
difficult to select (goal NAV-, see Section 3.3.6); changing the structure of the menu might hamper natural, intrinsic organization of application functionality (goal EASY-, see Section 3.3.1.1); moving a ring away from the center also increases the distances to the items (goal NAV-, see Section 3.3.2); and item overlapping decreases effective size of the items (goal NAV-, see Section 3.3.6) and possibly degrades their legibility (goal EASY-, see Section 3.3.7).



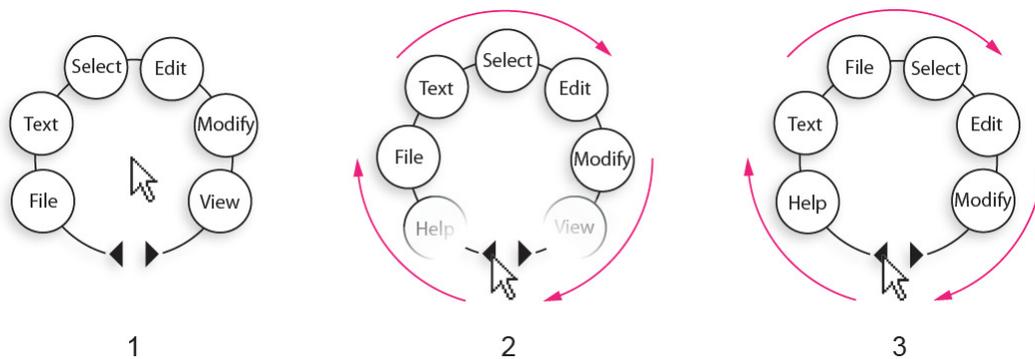
**Figure 3.15. Two approaches to accommodate more items on a ring. The first approach (a) increases the radius of a ring, which also increases its circumference. The cost is the slightly increased distances to the items. The second approach (b) uses item overlapping to pack the items more tightly. The costs are the decreased effective size of the items and possibly hindered legibility.**

If items overlap (see Figure 3.15b), hovering over an item might bring it to the front for better legibility (goal EASY-, see Section 3.3.7). Also the surrounding items can be drawn aside to visually separate the focused item. Alternatively, a fish eye technique [Furnas, 1986; Bederson 2000] can be used to allocate more space to the items close to the cursor (see Figure 3.16).

We also came up with a fourth remedy for dealing with the limited space: *ring scrolling*. It works as follows. If a ring cannot accommodate the desired number of items, it does not close entirely. The user can use a mouse roll, graphical arrows or other scrolling mechanisms to rotate the content of the ring, clockwise or counter clockwise. The items moving towards an open part of the ring disappear and new items appear on the other side (see Figure 3.17). This feature can be thought of as a metaphor of a screw. Only the part at the edge of the surface is visible. But the user can drive a screw inside or outside the surface to change the visible part. In the case of ring scrolling, the cost is the time needed for scrolling (goal NAV-).



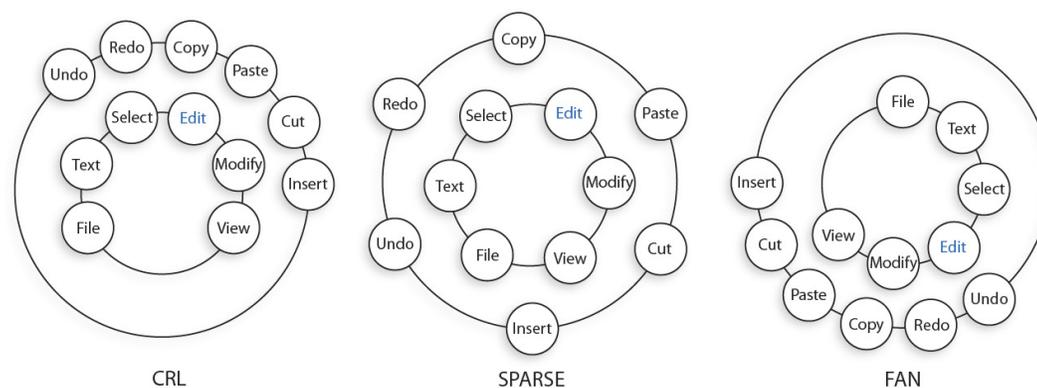
**Figure 3.16.** Three ways of handling item overlapping: a) bring a targeted item to the front; b) draw aside items surrounding a targeted item; c) use the fish eye technique, in which the space is distorted and the items around the cursor are allocated more space than the items farther away from the cursor.



**Figure 3.17.** Ring scrolling. Using a mouse roll, graphical arrows, or other scrolling mechanisms, a user can rotate the content of the ring. Some existing items disappear in the open part of the ring (i.e., at the bottom of the ring), and some new items appear on the other side. In the figure, a user clicks on the left arrow which results in a clockwise rotation of the menu content, as indicated by the arc arrows. The “View” item disappears on the right hand side and the “Help” item appears on the left hand side.

### 3.4.2. Item arrangement

If concentric rings are used to arrange menu levels, there is still the question of how to arrange items within the rings. Figure 3.18 shows three alternatives, all preventing overlaps (goal EASY+, GUIDE+, SEARCH+, NAV+, see Section 3.3.4).



**Figure 3.18.** Three spatial arrangements of menu items on the rings. The item “Edit” is selected on the first ring and therefore the second ring shows its children.

The Compact Radial Layout (CRL) arrangement packs items tightly around their parents. The SPARSE arrangement distributes items evenly within a ring similarly to pie menus. The FAN arrangement places the first item from a submenu on the next outer ring in the closest position to the parent item (i.e., the item “Edit”). All further menu items are placed clockwise around the outer ring.

The three arrangements present different trade-offs between distances, spacing, and discrimination methods. The CRL arrangement decreases the navigation distances the most but at the cost of decreased spacing and angular differences between the items. By contrast, the SPARSE arrangement increases the spacing and angular differences but increases the distances.

Different arrangements might do a better or worse job when it comes to guiding users in their visual search. The FAN arrangement, for example, clearly indicates the beginning of the subsequent level since the first item lays just next to the parent item and the rest of the items span clockwise. The same cannot be said about the SPARSE arrangement. The question of the effects of arrangement on visual search is open and needs experimentation. Addressing this issue is one of the goals of the experiment presented in Chapter 6.

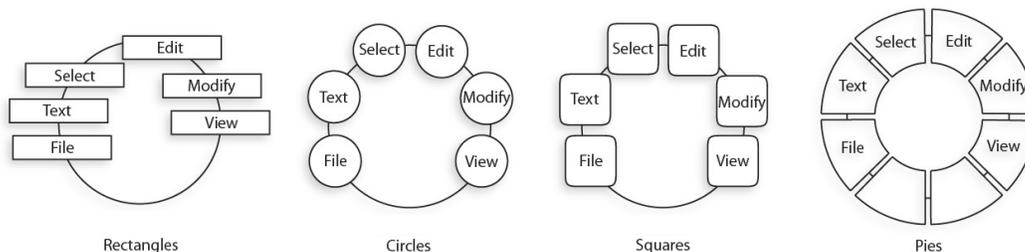
In Section 3.3.5 we discussed the effects of item ordering on navigation and search performance. Be it SPARSE, FAN, or CRL arrangement, the question of how to order the items (i.e., which visual item to assign to which command) for optimal access also remains unanswered. Previous research on visual search and item ordering focused

predominantly on linear menus. Unfortunately, those findings cannot be readily generalized to radial menus because the layout differences are considerable and search is known to be sensitive to small differences in interface design [Hornof, 2001; Pirolli et. al., 2003; Everett and Byrne, 2004; Halverson and Hornof, 2004]. Rings could be searched clockwise, counter-clockwise, top-to-bottom, bottom-to-top etc. The search could start from an item closest to the parent item or from an item on any end of an arc. Since the effects of ordering can be considerable they will be experimentally explored in Chapter 7.

We chose the CRL arrangement for our final design since it decreases the distances the most. Therefore we expect better navigation and search performance (goal NAV+, SEARCH+, see Section 3.3.2). We decided, however, to add some spacing between the items to minimize the consequences of under- and overshooting errors (goal NAV+, see Section 3.3.4).

### 3.4.3. Item representation

Figure 3.19 presents four alternative item representations. As explained previously in Section 3.3.6, the size of a rectangular item is small when approached from the top or from the bottom (goal NAV-). Rectangles also occupy a considerable amount of horizontal space and thus would require an increased radius of the subsequent rings in order to avoid overlaps (goal SPACE-).

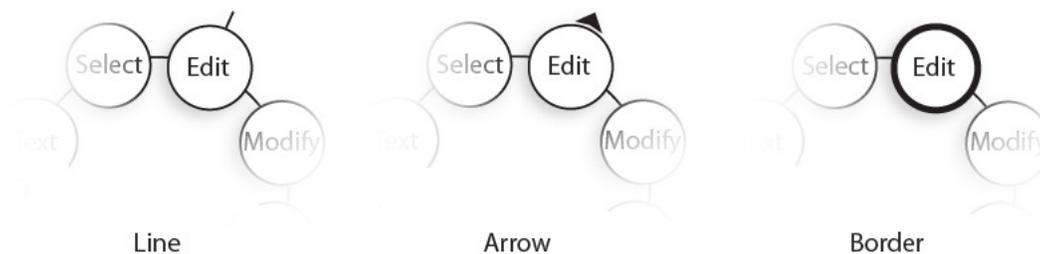


**Figure 3.19. Four item representations: rectangles, circles, squares, and pies.**

Circles, squares, and pies are preferred for our design as their sizes are comparatively larger than those of the rectangles (goal NAV+, see Section 3.3.6). Circles always have the same size regardless of the angle of approach. These shapes are also more suitable for icons than thin rectangles because they do not overly restrict the vertical space (goal EASY+, GUIDE+, see Section 3.3.7). As mentioned in Section

3.4.2, we introduce spacing between the items to decrease chances of selecting an adjacent item upon under- and overshooting errors (goal NAV+).

For our final design we chose to use circles. To distinguish between structural and functional items, we mark all structural items (i.e., parent items) with a line pointing outward (goal EASY+, GUIDE+, see Section 3.3.7) (see Figure 3.20). Two other alternatives, arrows and thick borders, are shown in Figure 3.20.



**Figure 3.20. Three ways of representing parent items. A line and arrow pointing outward might better communicate that an item leads somewhere (i.e., to another level) than the thicker border.**

In a cascading menu, thin rectangular boxes seem to be well suited to labels of various lengths. However, the same cannot be said about circular or square items. There is less horizontal space (goal EASY-, GUIDE-, see Section 3.3.7). The item size can be increased to provide more space for longer labels, but the costs are the increased navigation distances and screen consumption (goal NAV-, SPACE-, see Section 3.3.2).

To accommodate long labels we decided on three things. First, labels need not be contained inside the shape of the item but can run through its borders. Border crossings might impact label legibility. This can be handled by reducing the contrast between items (see Figure 3.21). Second, labels can be split into multiple lines at word borders (see Figure 3.21). This seemed to be a good idea because uniform shapes, such as circles and squares, provide more vertical space than thin rectangles. Third, the spacing between the items can be increased to prevent label overlaps.

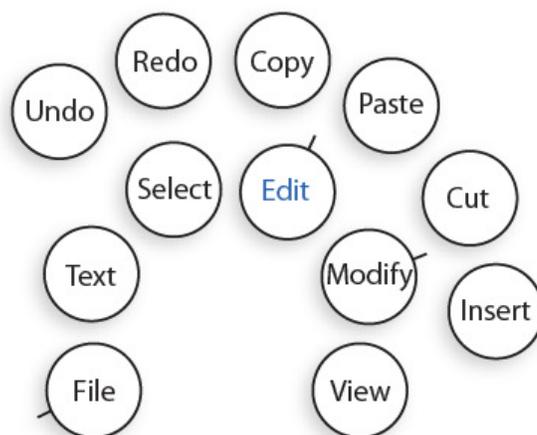


**Figure 3.21.** To accommodate long labels, they can run across item borders and span multiple lines. The items are dimmed to diminish interference with the label and prevent legibility problems.

To determine the exact sizes of circular and square items we did not match their surface area with the area of representative cascading menu item. Rather, we experimented with different configurations and decided to use items of diameter/diagonal in the range of 45-55 pixels. This size provided enough space to accommodate three- and four-word labels – eventually broken up onto two or three lines—while avoiding overlaps.

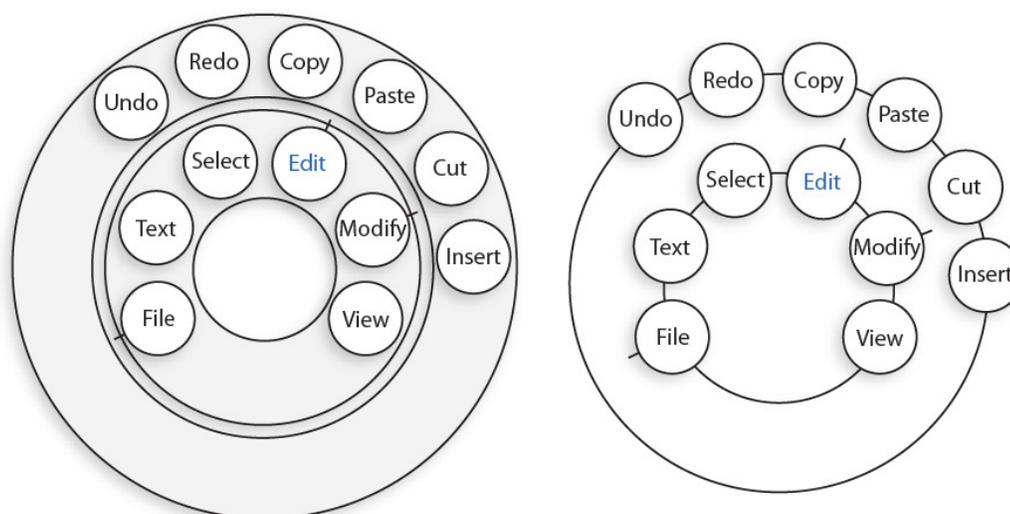
#### **3.4.4. Level representation**

Menu levels are not directly interacted with. Often they do not have corresponding visual representation, such as in cascading menus. Figure 3.22 shows our radial menu in which levels do not have visual representations. However, one re-occurring comment from informal interviews was that when more levels are displayed the entire menu looks like a blob of items without any particular structure (goal EASY-, GUIDE-, see Section 3.3.7).



**Figure 3.22.** A Compact Radial Layout menu without visual representation of levels (i.e., rings). It might be difficult to perceive the hierarchical structure of the menu content.

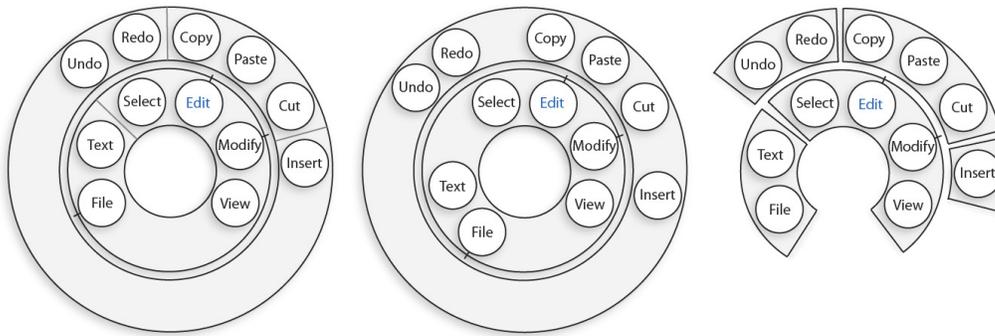
In the case of cascading menus, although there is not a dedicated visual representation of a level, the outer edges of the items belonging to the same level create a perceptual box. And since these boxes are separated spatially they strongly communicate the structure of the menu (goal EASY+, GUIDE+, see Section 3.3.7). The same cannot be said about spatially separated circular items running across an invisible arc. For this reason, we decided to represent levels visually with rings to preserve sense of menu structure (see Figure 3.23).



**Figure 3.23.** Compact Radial Layout menus with visual representations of levels (i.e., rings). The representations aid in the perception of the hierarchical structure of menu content.

### **3.4.5. Item grouping**

To support item grouping we came up with three different approaches (see Figure 3.24). The first one uses line bars in a similar way to cascading menus. The second approach relies on spatial proximity: there is more spacing between the groups than between the items within the same group. This could be viewed less as visual representation technique and more as a layout technique since it is based on altering distances. The third approach divides a level representation (i.e., a ring) into a set of arcs, each representing one group.



**Figure 3.24.** Three alternative approaches to group menu items: a) separating groups with line bars, b) separating groups with extra space (spatial groups), c) separating groups into disjoint arcs.

### 3.4.6. Menu navigation

The proposed design supports the concept of unfolding menus (goal EASY+, GUIDE+, SPACE+, see Section 3.3.8.2). A user is initially presented only with the first level (i.e., one ring) of the menu hierarchy and then reveals further levels by selecting parent items. Selecting a non-parent item closes the menu and invokes the associated command.

In one of the prototypes we explored the idea of aligning all selected items into a straight line to more clearly represent the current navigation path. However, such alignment requires repositioning items (i.e., with a rotating ring) which might hamper development of a mental model for the menu (goal N2E-, see Section 2.8.1, Section 3.2.3, and Section 3.3.5). For this reason we decided to not change the locations of menu items within rings to ensure positional constancy of the layout.

The design supports navigation mechanisms and cues described in Section 3.3.8.2: item previewing, level previewing, and path highlighting. Item previewing and path highlighting use the colour blue which is similar to common cascading menus. Level previewing (see Section 3.3.8.2) dims the previewed level. In our design, all previous levels and their items are visible and can be selected. Consequently, users can jump to any part of the context directly (goal EASY+, NAV+, see Section 3.3.8.1). This supports exploratory search behaviour but also eliminates the need for a dedicated mechanism that allows returning to previous levels.

### **3.4.6.1. Selection method**

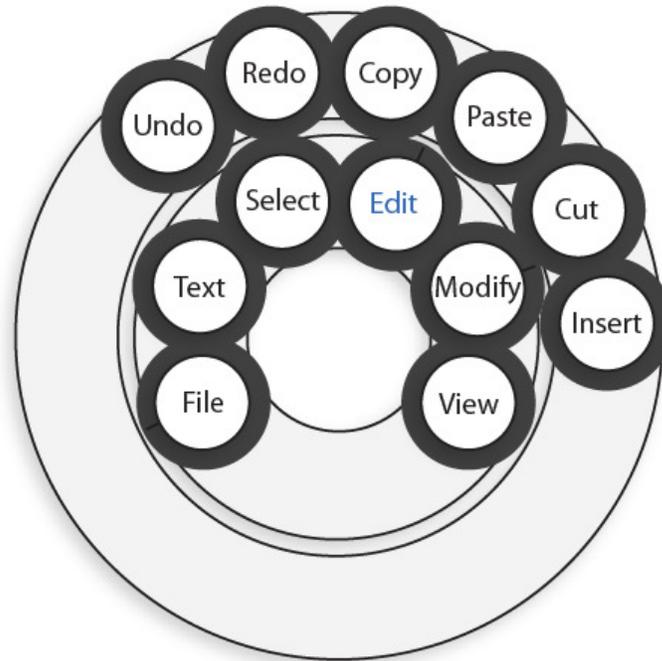
Item selection requires three basic actions from a user. First, users need to invoke the menu if it is not visible. We refer to this as *menu invocation*. Second, users select parent items to access their sub-menus. We refer to this as *nonterminating selection*. Third, users select the desired menu item. This item does not have a sub-menu and can be on any level of the menu hierarchy. We refer to this as *terminating selection*.

These three basic actions can be accomplished using different interaction methods such as Point-and-Click, Drag-and-Drop, and dwelling (see Section 3.3.9). We chose Point-and-Click for both, terminating and nonterminating selections, because it seems to be the most natural way of directly selecting screen objects (goal EASY+, see Section 3.3.9) and because it is faster than other alternatives (goal NAV+, see Section 3.3.9). We also chose to avoid dwelling in favour of button presses in order to not restrict navigation trajectories (goal NAV+, see Section 3.3.9). For menu invocation we decided to mimic context menus (goal EASY+, see Section 3.2.1). For example, if a mouse is used, typically a right mouse button invokes a context menu.

### **3.4.6.2. Aborting selection**

A menu needs to provide a mechanism for aborting the current selection. That is, the user needs some means of cancelling the selection while it is in progress. When it comes to Point-and-Click, the common approach is to perform a click outside the menu. In that case, the menu disappears and none of the commands is selected. We decided to use this familiar mechanism (goal EASY+, see Section 3.2.1). However, this decision impacts costs of under- and overshooting errors. When they occur, instead of selecting a desired item, the menu disappears because a click outside the menu is interpreted as “abort the selection”. Thus the cost of under- and overshooting errors can be considered high, especially if a mistake is small (goal NAV-, see Section 3.3.4). Although our design explicitly tries to mitigate under- and overshooting errors by using increased item sizes, uniform shapes, and additional spacing between the items, we cannot assume that these errors will never occur. To this end, we decided to introduce a passive *menu area*. This area embraces the menu and is supposed to cover those locations where under- and overshooting is more likely to occur (see Figure 3.25). If the user clicks in this area, nothing happens – none of the items is selected and the menu does not disappear (goal NAV+, see Section 3.3.4). We expect that users trying to abort the current selection will not try to do so by clicking between the tightly

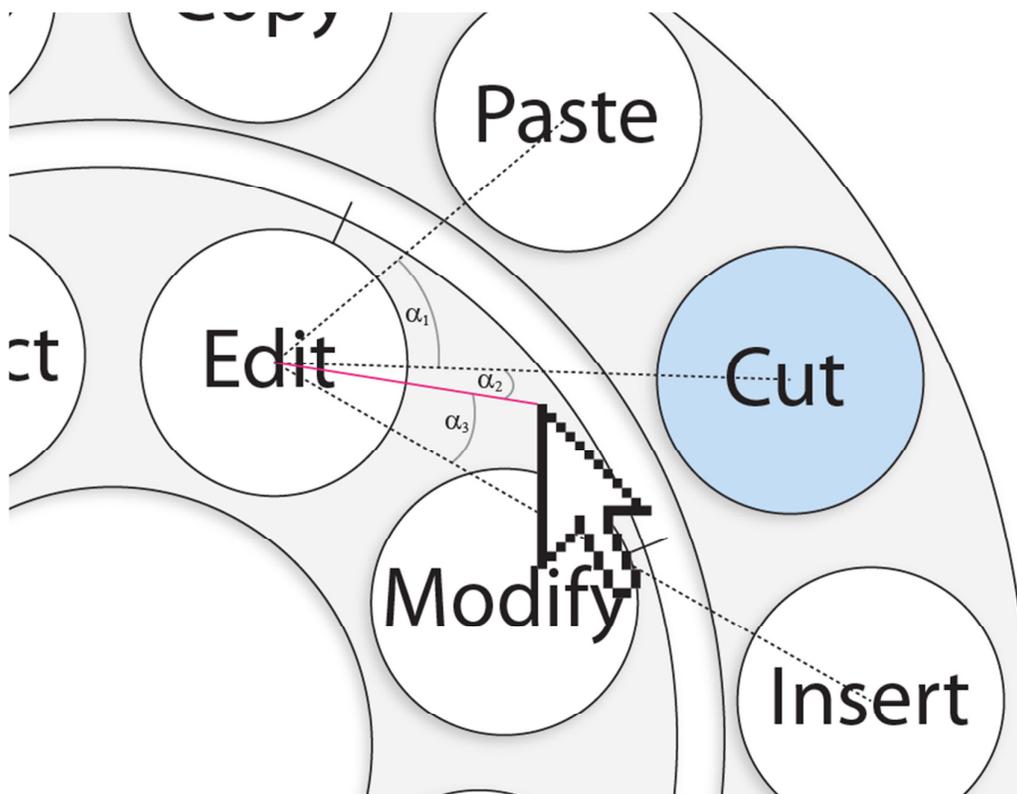
packed items. Rather, to avoid accidental item selection we expect them to issue a click outside the passive menu area.



**Figure 3.25.** The passive menu area (dark circles) covers regions where under- and overshooting errors are more likely to occur. Clicking on the passive menu area does not result in menu closing.

The size of the menu area was arbitrarily set to 5 pixels based on our experience with early prototypes.

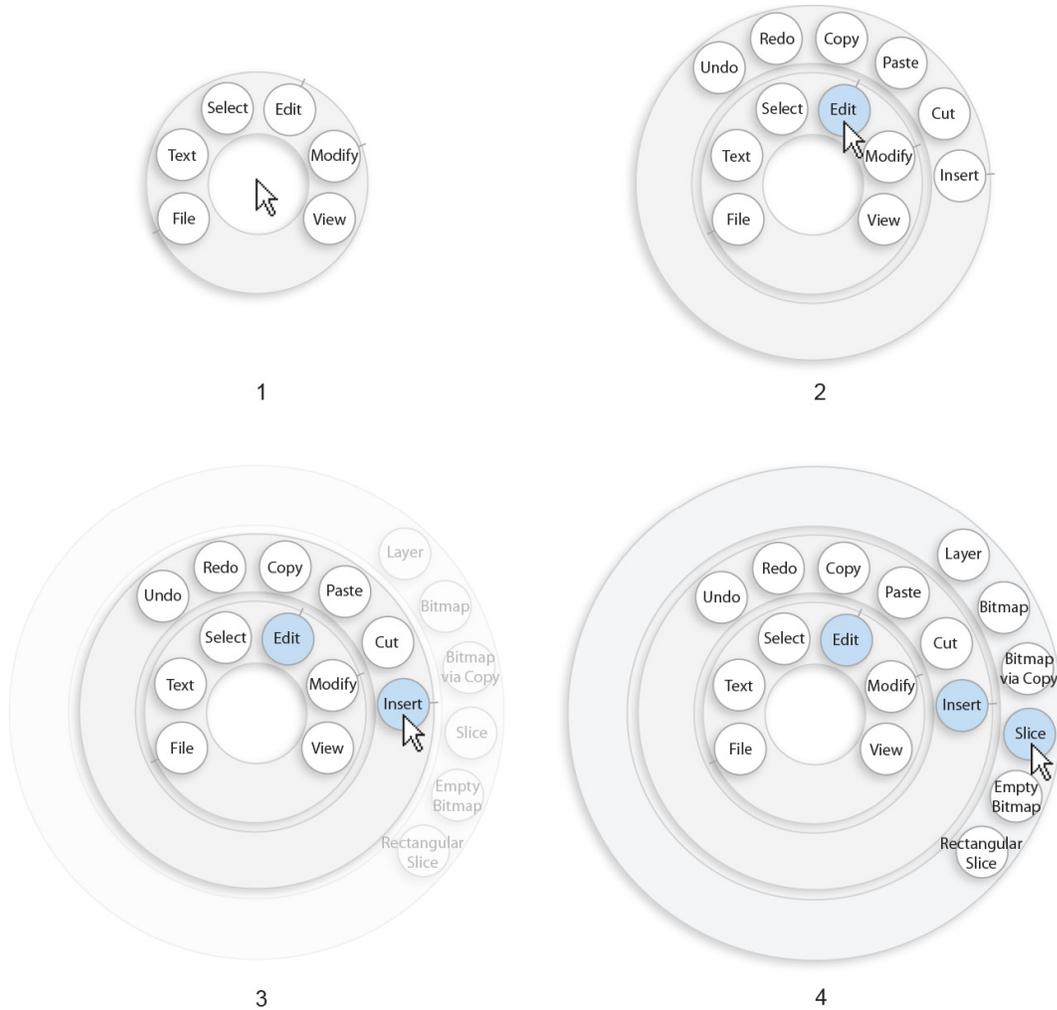
In Section 3.3.3 we discussed that the angle-based discrimination method might allow the prediction of which item a user intends to select. The prediction can be based on the direction of the cursor movement, what is currently below the cursor, and the proximity of menu items (see Figure 3.26). We suggested that the predictions could be coupled with appropriate techniques (for examples see Section 3.3.3) which might enable faster selections, not necessarily just by terminating inside a desired item but well in advance (goal NAV+, see Section 3.3.2 and Section 3.3.3). Figure 3.26 shows a simple approach where a predicted item is highlighted before the cursor enters its spatial representation. If a user issues a click the item will be selected.



**Figure 3.26. Item prediction.** The  $\alpha_2$  angle is very small and it is considerably smaller than  $\alpha_1$  and  $\alpha_3$ . The cursor also is not over any menu item. The analysis of the cursor movement strongly indicates that the user intends to select the item “Cut”. Consequently, this item is highlighted as if the cursor was over the item already. The user can select the item with a mouse click.

### ***3.5. Summary of the design***

Figure 3.27 depicts the Compact Radial Layout (CRL) menu design, showing an example of a selection sequence. The levels of the menu hierarchy are represented as concentric rings. The first ring corresponds to the first level of the menu hierarchy. Subsequent rings, surrounding the previous ones, correspond to deeper levels of the menu hierarchy. Each such ring represents a submenu of a parent item selected from a previous ring. The menu items are represented by circles tightly packed within the ring. All parent items are additionally marked with a line coming out of the item.



**Figure 3.27.** An example of a selection from a Compact Radial Layout menu. A user first opens the menu (1). Then he selects the item “Edit” with a mouse click (2). Next, the user hovers over the item “Insert” which brings up a sub-menu preview (3). The user sees a desired item in the preview (i.e., the item “Slice”). Therefore, he selects the item “Insert” and moves to the item “Slice” to issue the final selection (4). Note how the selected items and the item currently under the cursor are highlighted.

The interaction with a CRL menu works as follows: the user invokes the menu with a right mouse button click. The first ring surrounding the cursor appears. The user then points to a desired item and selects it with a left mouse click. If the item has no submenu, the command corresponding to that item is invoked and all the rings disappear. If the item *does* have a submenu, the next outermost ring is displayed. Similarly, any subsequent rings will appear if the user selects a parent item located on

the outermost ring. The user can select from any visible items. If a parent item on an inner ring is selected, the now unnecessary outer rings disappear automatically and a new outer ring appears. The selection path is highlighted as well as an item under the mouse cursor. If the cursor is placed over a parent item, its submenu is previewed using dimmed colours.

### ***3.6. Roadmap for experiments***

As explained in Section 3.2.5, the design part of this thesis used the design goals to make the best design decisions based on the previous findings and our subjective judgments. However, to support our central hypothesis, the goal NAV needs to be investigated thoroughly in rigorous experiments. The following measures will be employed in this investigation:

- 1) **Task completion time** in a menu navigation task (Chapter 4).
- 2) **Pointing time** component in a menu selection task (Chapter 6).
- 3) **Expert menu performance** of menu selection (Chapter 6).
- 4) **Speed/Accuracy Trade-Off** in a menu navigation task (Chapter 4).
- 5) **Navigation errors** in a menu navigation task (Chapter 5).

The menu presented in this chapter employs a radial layout which is considerably different than the linear layout. For this reason, the goal SEARCH also needs experimentation in order to understand the effects of layout on search performance. This will be achieved by employing the following measures:

- 1) **Visual Search time** component in a menu selection task (Chapter 6).
- 2) **Novice menu performance** of menu selection (Chapter 6).

The performance of NAV and SEARCH might highly depend on item ordering (see Section 3.3.5). To this end, the Chapter 7 will investigate patterns of how users search and point to items on the rings. Investigations of NAV and SEARCH will also focus on testing how different arrangements presented in Section 3.4.2 influence the above measures. The experiments will perform the tests for different menu sizes and different target levels. Finally, apart from the objective measures we will also employ the subjective measures to understand the user perception of the performance, error rates, and user satisfaction.

The Chapter 6 will assess experimentally the goal SPACE by measuring the number of pixels consumed by each tested design and the number of pixels consumed by a bounding box and a bounding circle.

## ***4. Speed/Accuracy Trade-Off in Hierarchical Menu Navigation***

This chapter presents a study that focuses on navigation times in two menus: a linear cascading menu and the CRL menu. The study employs quantitative and qualitative measures. The results are used in a two-fold way. First the results are used to compare both menus. The comparison shows that the CRL menu enables faster navigation than the CS menu in variety of conditions. Participants also perceive the CRL menu as being faster. Second, the results are used to build a performance model based on pointing laws which would provide detailed explanations regarding tested designs. In particular, such a model can assess speed/accuracy strategy employed by the participants.

Applying pointing laws to model single pointing tasks has been widely explored in previous work. However, in our context we need to model not a single pointing tasks but rather sequences of pointing tasks because a selection from an unfolding hierarchical menu employs a sequence of selections, one for each menu level. This chapter argues that existing models of such selections have methodological problems preventing them from providing meaningful interpretations.

The Sequential Pointing Model (SPM) is proposed in response to these problems. The model is built on the data collected in the experiment. The modelling results demonstrate that compared to the CS menu, the CRL menu decreases difficulty of structurally equivalent tasks and allows users for faster and less precise movements.

The rest of the chapter is structured as follows. Section 4.1 presents background material focusing on pointing laws and their application to menu modelling. Section 4.2 presents the Sequential Pointing Model of menu navigation. Section 4.3 describes the data collection experiment. Section 4.4 compares the performance and subjective perception of the tested designs. Section **Error! Reference source not found.** builds the SPM model and discusses the insights it provides. Section 4.6 and Section 4.7 present discussion and conclusions.

## **4.1. Background**

The background is divided between two sections. The first section discusses Fitts' law and the steering law. The second section reviews work on menu modelling with a special focus on modelling menu navigation.

### **4.1.1. Pointing laws**

#### **4.1.1.1. The Fitts' law and the steering law**

Fitts' law and the steering law have been particularly successful in the HCI context [Card et. al., 1978; MacKenzie, 1992a; MacKenzie, 1992b; Accot and Zhai, 1997; MacKenzie and Soukoreff, 2002; Gutwin and Skopik, 2003; Soukoreff and MacKenzie, 2004]<sup>3</sup>. This success stems from the simplicity of these pointing laws, their robustness, and the importance and frequency of pointing tasks in the use of computers.

Fitts' law [Fitts, 1954] models rapid, aimed movements and is derived by analogy from information theory. A single movement task is represented by the Index of Difficulty (ID, a logarithmic expression shown in Equation 1), measured in bits, which carries information about how difficult a particular pointing task is. The ID is a function of the distance to an item ( $d$ ) and its size ( $w$ ). To accomplish a task, the necessary information bits need to be transmitted through the user's motor channel. And because this channel has limited capacity, the transmission takes time. The exact formula of the Fitts' law can be expressed in various ways. The Welford formulation is used in this chapter (see Equation 4.1).

$$\text{Movement Time} = a + b \log_2 \left( \frac{d}{w} + 0.5 \right) \text{ ms} \quad (4.1)$$

The steering law [Accot and Zhai, 1997] models trajectory-constrained movements where the user needs to move a cursor within a narrow tunnel of width  $w$  and length  $d$  (see Equation 4.2).

---

<sup>3</sup> See also International Journal Human-Computer Studies – Special issue: Fitts law 50 years later: Applications and contributions from human-computer interaction, 61 (6), 2004.

$$\text{Movement Time} = a + b \frac{d}{w} \quad (4.2)$$

Note that the ID component of the steering law is not logarithmic like in the case of the Fitts' law but linear – denoting the increased difficulty of trajectory-constrained tasks.

Many models are void of physical interpretation. A key feature of the pointing laws is the correspondence of their parameters to physical properties underlying the movement task. An interpretation is offered for each term in the equation [MacKenzie, 1992a].

The slope is typically interpreted as the rate of transmitting information through the human motor channel; the time required to transmit one bit of information. It is the *information* aspect of input, a portion of movement time attributed to the change of ID [Soukoreff and MacKenzie, 2004; Zhai, 2004]. The slope identifies the speed/accuracy trade-off for a given pointing system (e.g., a pointing device or interface technique). Extensive prior research shows that values for the human motor processing rate using a mouse device range from about 90 ms/bit to 300 ms/bit [Card et. al., 1978; Card et. al., 1983; MacKenzie, 1992a; Byrne, 2001; Hornof, 2001].

The intercept, on the other hand, represents the *non-information* aspect of input – the one off cost of a movement task independent of the ID. In the HCI context, the intercept is particularly sensitive to additive factors, and as such can be used to estimate their size. Several interpretations have been offered as to what additive factors are captured in the intercept (adapted from [Soukoreff and MacKenzie, 2004]):

- (1) The time required to make a movement of zero distance [MacKenzie, 1992b, p. 142].
- (2) Navigation components such as time required for button presses [MacKenzie, 1992a, p. 109], the application of pressure, or dwell time [Fitts and Radford, 1966, p. 476].
- (3) Unavoidable delay in the psychomotor system [Fitts and Radford, 1966, p. 476], uncontrollable muscle activity at the beginning or end of the movement task [MacKenzie, 1992a, p. 109].
- (4) Reaction time [Fitts and Peterson, 1964, p. 111].

Some of the additive factors are result of a pointing system; others can be attributed to human perceptual, cognitive, and motor characteristics. Consequently, the exact extent of the additive factors may vary for different pointing systems and tasks.

#### ***4.1.1.2. The advantages of using pointing laws***

Modelling pointing tasks using Fitts' law and the steering law provides insights beyond those offered by traditional comparative studies. The main advantages of such modelling include:

- **Assessing task difficulty.** The IDs allow quantitative assessment of task difficulties. For example, two different menu designs can be compared by calculating the difficulties of structurally-equivalent tasks. Note that this technique does not require the actual menus to be already implemented. Rather, it can provide feedback in the early design stages.
- **Meaningful model parameters.** As mentioned in the Section 4.1.1.1., the parameters of the pointing laws afford meaningful interpretation. The slope can reveal if tested pointing systems induce more or less accurate movements. The slope can be also compared against the reference values of human motor performance to test if a pointing system uses the full potential of the human motor capacity. The intercept can reveal differences in mental preparation needed for different designs, or the extent of other additive factors.
- **Test independence.** If pointing system A and B are compared by measuring the average time needed to point at targets, the results will depend on and be limited to the settings of the test (e.g., target size and distance). Pointing laws can convert such measures into parameters  $a$  and  $b$  (slope and intercept). And if done properly, the parameters will characterize the quality of the interaction technique, beyond the conditions used in the test. It is important to note, however, that the interpretation of slope and the intercept holds only over the range of IDs used when building the model.
- **Predictive power.** Models resulting in strong fit with the empirical data can serve as predictive models of a particular class of pointing systems. For a new pointing system the IDs of the pointing tasks can be calculated and then turned into predicted pointing times. Examples of such predictive models for menu performance include [Cockburn et. al., 2007] and [Ahlström et. al., 2010]. This advantage enables early feedback on interface design without running costly

experiments, and even before any implementation.

- **Cross study comparisons.** Test independence and meaningful model parameters enable cross-study comparisons. There are a number of difficulties for achieving this goal in practice. However, a considerable body of research focuses on this subject [MacKenzie, 1992a; Soukoreff and MacKenzie, 2004].

#### ***4.1.1.3. Building models for single movement tasks***

A task used extensively for modelling purposes is the Fitts' tapping task [Fitts, 1954]. Its advantage for model building stems from both its one-dimensional simplicity and its serial nature. As such, the task minimizes the contribution of other factors and allows for meaningful comparison of experimental conditions such as different pointing devices.

However, the Fitts' tapping task is not a particularly common example of user interaction with a computer. The HCI as a field is highly concerned with understanding usability and user behaviour for real interface designs. Consequently, in the HCI context, the pointing laws are often used to model tasks such as navigation in menus [Cockburn et. al., 2007; Ahlström et. al., 2010], navigation in visual hierarchies [Hornof, 2001], or target acquisition in 3D gaming environments [Looser et. al., 2005]. Application of the pointing laws to these tasks seems natural since they require a single movement task. These tasks typically also require responses such as a button press, application of pressure, or dwelling. Such responses, however, are associated with a one-off cost and thus are assumed to have an additive effect, contributing to the intercept of the regression line but not to the slope [MacKenzie, 1992a; Fitts and Radford, 1966]. Consequently, the meaning of the slope and the intercept is preserved as the information and non-information aspect of input respectively.

As a result, pointing laws applied in the HCI context to single selection tasks enable building meaningful models and learning about the differences regarding various visual structures, information layouts, interaction styles and a plethora of other design parameters. For example, the calculated IDs reveal how tested designs differ in terms of the difficulty of structurally equivalent tasks. The slope shows the speed/accuracy trade-off induced by a design. And the intercept informs about the mental preparation or other effects of additive factors.

#### **4.1.1.4. Building models for sequence of movement tasks**

Many real HCI navigation tasks go beyond single movement tasks (i.e., beyond single point-and-select). One particular type of such a multi-part task is navigation that requires a sequence of movement tasks where each movement task needs to terminate before the next one commences. Each movement task is completed with a select operation, which may reveal a new portion of the interface (e.g., a new sub-menu in a hierarchical menu, or a new dialog box).

The above type of navigation is common in computer use; examples include navigation in hierarchical menus, selecting multiple options in dialog boxes, and navigating through hierarchical index pages. The sequence of movement tasks, rather than each individual movement task, accomplishes a single logical user action such as invoking a menu command located in the third menu level.

To benefit from the insights offered by the pointing laws, designs requiring sequences of movement tasks need to be modelled in their entirety. Such modelling is the objective of this chapter.

One approach for achieving this objective is to first compute the ID of each individual movement task using Fitts' and steering terms and to then use either each ID or the sums of all terms of the same type as separate model components [Pastel, 2006]. This approach is similar to the one used for modelling combined steering-targeting tasks [Dennerlein et. al., 2000]. The problem, however, is that the result is not a single slope and an intercept but rather a set of slopes and an intercept. Each parameter represents the characteristics of one movement task rather than a tested design in its entirety. Such a model is relatively complex: it is difficult to interpret, summarize, and compare different designs, especially if they require different numbers of movement tasks or employ different types of movement tasks; and this makes it difficult to build more generic predictive models.

Another approach used in practice is to compute one compound ID representing the total difficulty of the entire sequential task [Ahlström, 2005; Ahlström et. al., 2006; Kulikov and Stuerzlinger, 2006; Ahlström et. al., 2010]. The compound ID is a sum of the IDs calculated for each separate movement task using a term of an appropriate law. The result is not different from a model for a single movement task: there is one slope and one intercept.

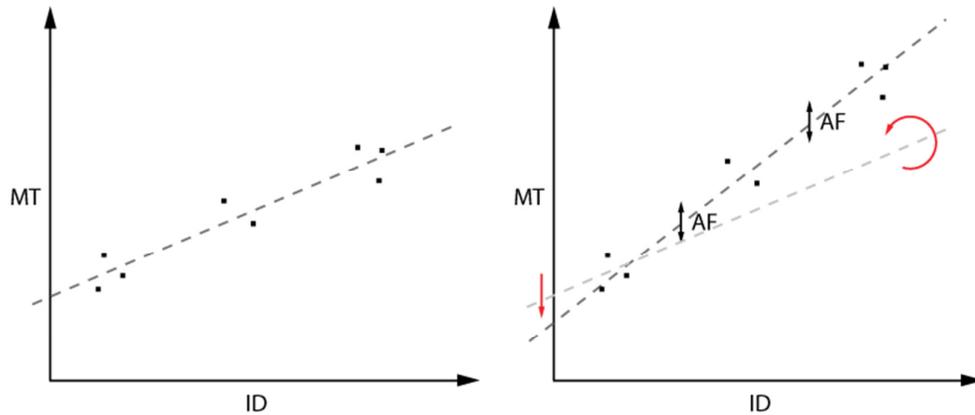
We believe that both approaches have methodological problems which might skew the meaning of the model parameters, prevent the benefits listed in Section 4.1.1.2, and lead to misinterpretations. This is further explained below.

Models based on either sequences of IDs or on the compound ID assume that additive factors occur only once for the entire sequential task and thus that the additive factors are associated only with a one-off cost represented by the intercept, just like in the case of modelling a single movement task (see Section 4.1.1.3). However, in practice, in the context of the computer use, each individual movement task in the sequential task is associated with some additive factors such as mental preparation, button press, or dwell (see Section 4.1.1.1). Consequently, the total effect of the additive factors is not a one-off cost but rather depends on the number of the individual movement tasks—which often varies across the trials. For example, the number of movement tasks in hierarchical menu selection depends on the level of the target item.

Tasks with a higher number of movement tasks take disproportionately more time because more additive factors are involved. Soukoreff and MacKenzie [Soukoreff and MacKenzie, 2004] refer to this problem as artificially inflated movement time. In consequence the regression line is tilted counter-clockwise, increasing the slope and pushing the intercept lower (see Figure 4.1). As a result, the model parameters lose their meaning as they mix the effects which normally would be attributed to the information and non-information aspects of the input. Such models cannot benefit from the advantages listed in Section 4.1.1.2.

This work argues that models of sequential navigation need to explicitly account for the additive factors associated with each movement task in order to preserve the meaning of model parameters and afford clear interpretations.

The chapter focuses on a case of hierarchical menu navigation as an example of a sequential navigation task. The next section presents work focusing on menu modelling, in particular, modelling menu navigation.



**Figure 4.1.** Effects of sub-selection additive factors on regression line characteristics. Both graphs show movement times for the tasks of the same total difficulty as measured by the ID. The tasks on the first graph require a single pointing task while the tasks on the second graph require one, two, or three pointing tasks. The imputed involvement of the additive factors which occur between each pointing task tilt the regression line counter-clockwise, pushing the intercept lower and increasing the slope.

#### **4.1.2. Menu modelling**

The menu models aim at various levels of abstraction in their description. The abstract models sacrifice detailed explanations to gain ease of use, while less abstract models provide detailed descriptions of where the user's time is spent but sacrifice ease of use or precision. Less abstract models typically include components which have clear correspondences to either human psychological characteristics or artifact design.

The level of abstraction helps in understanding the limits and benefits of the models as well as in better positioning the model built in this chapter between the existing ones.

##### **4.1.2.1. Novice and expert menu selection**

There are three components of menu selection: visual search, decision, and navigation [Cockburn et. al., 2007]. Novices search for an item and then navigate to it. Experts, on the other hand, know item locations so they do not have to search for them [Somberg, 1987; Kaptelinin, 1993]. They decide which item to select and then navigate to it [Landauer and Nachbar, 1985; Cockburn et. al., 2007]. If the menu is hierarchical, search and navigation (novices) or decision and navigation (experts) may be performed for each menu level.

Navigation refers to all motor actions that a user needs to perform when making a selection from a menu (e.g., moving the mouse pointer, dwelling, mouse clicks). Navigation is an important part of menu selection accounting for 20-50% the total selection times for both novices and experts [Samp and Decker, 2010].

Since the focus of this chapter is modelling navigation, the task and models built in this chapter do not employ visual search or decision making.

#### **4.1.2.2. Menu models not employing pointing laws**

A number of cognitive architectures, also known as theories of embodied cognition, have been used to model menu selections: EPIC [Hornof and Kieras, 1997], ACT-R [Anderson et al., 1997; Anderson et al., 1998], and ACT-R/PM [Byrne, 2001]. Cognitive architectures are designed to simulate human intelligence in a humanlike way [Newell and Card, 1985]. These models focus predominantly on visual search strategies in single level linear menus and do not explore navigation in menu hierarchies.

Card, Moran, and Newell provide the seminal work on user interface modelling [Card et. al., 1980; Card et. al., 1983]. The proposed GOMS/KLM model defines a set of operators which describe elements of user interacting with a computer. It can be applied to hierarchical menu navigation. Let us consider navigation in a cascading menu as an example.

First, the user needs to prepare mentally for the first movement task (i.e., for the first sub-selection) which is represented by the M operator = 1.35 sec. Then, the user points at the item: the P operator = 1.1 sec. And once over the item, there is the dwelling time after which a next sub-menu is posted:  $R(t) = 1/3$  sec. This sequence is then repeated for each subsequent movement task (i.e., for each sub-level selection). The only exception is that for the final sub-selection there is no dwelling since the mouse click is used to select the final item; so no  $R(t)$  is involved. The KLM model also includes three other operators: H (homing), K (keystroking), and D (drawing), but these are not applicable in this example. Below are the calculated times for the selection on the first and the second menu level.

$$\begin{aligned} \textit{Selection Time (first level)} &= M + P = 1.35 \textit{ sec} + 1.1 \textit{ sec} \\ &= 2.45 \textit{ sec} \end{aligned}$$

(3)

$$\begin{aligned} \textit{Selection Time (second level)} &= M + P + R(t) + M + P \\ &= 1.35 \textit{ sec} + 1.1 \textit{ sec} + 0.333 \textit{ sec} + 1.35 \textit{ sec} + 1.1 \textit{ sec} \\ &= 5.233 \textit{ sec} \end{aligned}$$

Comparing the KLM predictions with empirical results from previous work shows that the KLM predictions exceed the typical menu selection times by roughly 100% [Ahlström, 2005; Cockburn and Gin, 2006; Cockburn et. al., 2007; Tanvir et. al., 2008; Samp and Decker, 2010]. The M operator (1.35 sec) or P operator (1.1 sec) alone exceed the typical menu selection times on the first level. The disparity escalates with selections on the deeper levels. The data collected in this chapter also support this observation (see Section 4.4).

The use of 1.1 sec as the average time for pointing seems to be crude compared to the precision accessible through the pointing laws. Furthermore, some studies demonstrate that mental preparation of a movement task can be on the order of 0.24 sec [Bekkering et. al., 1994]. At the same time, the KLM model sets mental preparation to 1.35 sec (the M operator) because it needs to accommodate a broad range of user interface tasks, not only preparation for a movement task. Card et al. also mention that the P operator does not include mouse clicks and the K operator represents only the keystrokes. The KLM model cannot provide a detailed explanation of hierarchical menu navigation since that is not its objective. On the abstraction level scale, the KLM model can be considered as a low-level description of a user interacting with a computer in a variety of interface tasks. It sacrifices precision for specific tasks such as hierarchical menu navigation in order to gain generality and preserve the ease of application.

Lane et al. [Lane et al., 1993] used GOMS/KLM to predict selection times in hierarchical menus. The research question was whether the M operator (i.e. mental preparation of 1.35 sec) is necessary before each sub-selection or rather only before the first sub-selection. The study demonstrated that mental preparation depends on user expertise, is not binary, and may range from 0 to 1.35 sec. The selections were

performed with keyboard shortcuts (e.g., first letters of command names) so that the displayed menu did not have to be used. Therefore, the study was mainly concerned with human memory – namely, if shortcuts constitute one or several memory chunks. Importantly, however, the work of Lane et al. hints at the presence of possibly large effects of the additive factors associated with each menu sub-selection.

#### ***4.1.2.3. Menu models employing pointing laws***

Ahlström [Ahlström, 2005; Ahlström et. al., 2006] proposed modelling selection times in hierarchical cascading menu with an approach based on the compound ID. The author asserted that each navigation path in a cascading menu consists of a sequence of alternating horizontal and vertical movements. The compound ID was calculated as a sum of Fitts' IDs, representing vertical movements (i.e., within sub-menu movements), and steering IDs, representing horizontal movements (i.e., between sub-menus movements). The experiment employed a task where all the items to be selected were highlighted, thus the participants did not have to search for items nor decide which item to select next. Consequently, the experiment and modelling focused on navigation in hierarchical menus. Two cascading menu designs were used. The model produced good fit with the empirical data ( $R^2$  ranged from .904 to .976 depending on device and cascading menu design).

Cockburn et. al. [Cockburn et. al., 2007] proposed a model of menu performance based on three components accounting for Visual Search time, Decision time, and Pointing time (the SDP model). The Fitts' law was used to model pointing time. The objective of the SDP model is to predict novice performance, expert performance and the rate of transition from novice to expert. The paper included a study where model predictions were matched against the empirical data. The resulting fit was very good (within 2% of empirical data). This work, however, focused only on single level linear menus. Thus the navigation consisted only of single movement task. The authors proposed that the model could be extended to cascading menus (i.e., hierarchical linear menus) by using the Ahlström's compound ID for the pointing component. The effect of possible additive factors associated with sub-selections was not mentioned.

The proposal of Cockburn et. al. [Cockburn et. al., 2007] was realized in the extended version of the SDP model [Ahlström et. al., 2010]. The pointing component used the compound ID as proposed in [Ahlström, 2005]. The model fit was strong. But again there was no account for the additive factors associated with sub-selections.

The work of Ahlström et. al. [Ahlström, 2005; Ahlström et. al., 2006; Ahlström et. al., 2010] and Cockburn et. al. [Cockburn et. al., 2007] produce models which fit the empirical data very well. However, these models account for the sequence of movement tasks by using a compound ID; this has methodological problems as discussed in Section 4.1.1.4. Namely, the effects of the additive factors associated with each movement task (e.g., necessary dwelling time and reaction time) are not captured by a separate component but rather are mixed into the slope and intercept. Consequently, the meaning of the model parameters is skewed and they cannot be interpreted the same way as the parameters of the pointing laws. And for this reason, many of the benefits of using pointing laws listed in Section 4.1.1.2 cannot be expected. This is not clarified in the previous work and might lead to misinterpretations. It is important to note, however, that as with the KLM model, the detailed explanation of hierarchical menu navigation is not the objective of the SDP model and its extension. Rather, they can be considered as relatively abstract models whose objective is to provide accurate predictions of novice/expert performance while preserving simplicity of application.

#### **4.1.2.4. Summary**

Previous work on menu modelling did not focus on providing detailed explanations regarding menu navigation. It is not that navigation is of little importance and not worth investigating in greater detail. Quite the contrary, as discussed in Section 2.2.1, menu navigation is an important part of menu selection concerning both, novices and experts. Commensurate with this importance, many researchers have worked on aspects concerning menu navigation [Pastel, 2006; Samp and Decker, 2011] or contributed menu designs that explicitly aimed at improving navigation [Kobayashi and Igarashi, 2003; Ahlström, 2005; Ahlström et. al., 2006; Cockburn and Gin, 2006; Cockburn et. al., 2007; Tanvir et. al., 2008]. As a result, there are many differences in how various menus are navigated. Some menus employ a point-and-click interaction style [Samp and Decker, 2010], others a dragging interaction style [Kurtenbach and Buxton, 1993]. Some menus restrict navigation trajectory when moving between the sub-menus, others do not [Cockburn and Gin, 2006; Tanvir et. al., 2008]. Some menus make the navigation to frequently selected items easier by dynamically increasing their sizes [Cockburn et. al., 2007]. Some menus facilitate navigation between the sub-menus; for example, by attracting the cursor towards an open submenu [Ahlström, 2005] or by opening a sub-menu faster if the cursor moves towards it [Kobayashi and Igarashi, 2003]. These are only few examples. The importance of menu navigation and

the plethora of menu navigation techniques make it important to develop models enabling better understanding of the actual advantages and disadvantages of the different navigation designs.

For this reason, the example of menu navigation used in this chapter is useful not only for investigating methodological issues related to applying pointing laws to sequences of movement tasks, but is also practical for menu research and menu design.

The objective of the chapter is to build a model which is less abstract than the SDP model and more precise than the GOMS/KLM model but with the limited scope of navigation in hierarchical menus. Less abstraction and more precision is sought through the use of pointing laws. Moreover, in contrast to the previous work the model is not intended as a predictor but rather as a technique of assessing navigation characteristics of chosen menu designs.

## ***4.2. Sequential Pointing Model***

This section presents two models of menu navigation. The first model uses the approach presented in the previous work; that is, an approach based on the compound ID [Ahlström, 2005; Ahlström et. al., 2006; Ahlström et. al., 2010, Kulikov and Stuerzlinger, 2006]. This model does not account for the additive factors on the sub-selection level. Further, a Sequential Pointing Model is proposed which adds a component explicitly accounting for the additive factors on sub-selection level.

### ***4.2.1. Compound ID model***

The model based on the compound ID has the following formulation:

$$\text{Movement Time} = a + b ID_T \quad (4)$$

The  $ID_T$  represents the total index of difficulty. It is calculated differently for the CS and the CRL menu.

#### ***4.2.1.1. The CS menu***

For the CS menu, the  $ID_T$  is calculated for each trial in the following manner [Ahlström, 2005]:

$$ID_T = ID_V + ID_H \quad (5)$$

where  $ID_V$  represents the total difficulty of all vertical tasks and  $ID_H$  represents the total difficulty of all horizontal tasks needed to select a menu item from its sub-menu.

Vertical tasks are modeled with the Fitts' law thus the  $ID_V$  is calculated in the following manner:

$$ID_V = \sum_{j=1}^m \log_2 \left( \frac{d_j}{w_j} + 0.5 \right) \quad (6)$$

where  $d_j$  and  $w_j$  is the distance to and the size of the target item in the  $j^{th}$  sub-menu. The target item is located in the  $m^{th}$  sub-menu. In case of the CS menu, the size of a target item equals its height since it must be approached vertically from the top and the distance is measured to the item's center.

Horizontal tasks are modeled with the steering law thus the  $ID_H$  is calculated in the following manner:

$$ID_V = \sum_{j=1}^{m-1} \frac{0.5w_j}{h} \quad (7)$$

where  $w_j$  is the width of the  $j^{th}$  sub-menu. The length of a steering task is assumed to be a half of the corresponding sub-menu width [Ahlström, 2005].

#### **4.2.1.2. The CRL menu**

For the CRL menu the same equations cannot be used as for the CS menu since issuing a selection in the CRL menu does not have a task which could be viewed as a steering task. On each level the user has to point and click on a desired menu item. Such a task can be viewed as an independent Fitts' task, and selecting a menu item on a deeper level as a sequence of Fitts' tasks. Consequently, for the CRL menu the  $ID_T$  is calculated as a sum of the Fitts' IDs rather than using both the Fitts' and steering IDs:

$$ID_T = \sum_{j=1}^m \log_2 \left( \frac{d_j}{w_j} + 0.5 \right) \quad (8)$$

where  $d_j$  and  $w_j$  represent the distance and the size of the target item in the  $j^{th}$  sub-menu. The target item is located in the  $m^{th}$  sub-menu. The size of the circular menu item equals its diameter – this is result of measuring the size along the line of approach as recommended in [MacKenzie, 1992a]. The distances are measured between the centers of the circles.

#### **4.2.2. The Sequential Pointing Model (SPM)**

This chapter proposes an extended model which includes a component explicitly accounting for the additive factors associated with sub-selections:

$$Movement\ Time = a + bID_T + cL \quad (9)$$

The IDs for the CS menu and the CRL menu are calculated the same way as for the compound ID model. The  $L$  indicates the level where the target item is located. It expresses how many times a user has to move from one sub-menu (level) to another in order to select the target item. Consequently, it equals the number of times the additive factors associated with the sub-selections occur.

The  $L$  is 0-based because the intercept of the model (i.e., parameter  $a$ ) captures the additive factors associated with the final sub-selection. Therefore, for a target item on the first level  $L = 0$ , for a target item on the second level  $L = 1$ , and so on.

The parameters  $a$  and  $c$  capture the size of the additive factors associated with the final and all the previous sub-selections respectively. Separation of the additive factors associated with the final sub-selection and all the previous sub-selections is both important and practical since they often differ. For example, the CS menu employs dwell for all sub-selections excluding the final selection which is made with a mouse click.

The name of the model – the sequential pointing model – stresses its emphasis on a sequence of pointing tasks rather than on single complex pointing task.

### **4.3. Experiment**

The experiment presented in this section is designed to collect navigation data in order to build the Sequential Pointing Model presented in Section 4.2.2.

#### **4.3.1. Menus**

Two hierarchical menus are used in this experiment. The first menu is the cascading menu (CS). It was chosen because it is the most generic menu deployed in many software products. The second menu is the CRL menu. The implemented CS menu was equivalent to those found in contemporary applications such as Microsoft® Word. In short, a mouse click was used to open the menu and to select the final item while dwelling over a parent item for 1/3 seconds opened a sub-menu. Moving between the sub-menus required steering: the sub-menus did not disappear immediately upon steering errors but after a short delay, and the item under the cursor as well as all parent items with open sub-menus were highlighted blue.

The implemented CRL menu was equivalent to that presented in Chapter 3. In short, a mouse click was used to open a menu, select a parent item, and select the final item. The items were represented by circles, the levels by concentric rings. The innermost ring corresponded to the first level and again the item under the cursor was highlighted blue. Figure 4.2 shows examples of selection sequences for both menus.

To determine the size of the items in the CS menu we analysed several menus of widely used applications and found that widths varied significantly. The size of 215×19 pixels (area = 4085 pixels) was chosen for the CS menu because this size could accommodate three word labels. To determine the size of the items (i.e., circles) for the CRL menu the area of the circular items was not chosen to match the area of the representative cascading menu item. Rather, the size for the circles was determined experimentally such that the circles provided enough space to also accommodate three-word labels—eventually broken up onto two or three lines—while avoiding overlaps. The diameters for the CRL menu items were 44 pixels (area = 1521 pixels), 48 pixels (area = 1810 pixels), and 52 pixels (area = 2124 pixels) for the first, second and third level respectively. For the experiment the animations in both menus were disabled.

### **4.3.2. Measures**

The study included quantitative and qualitative measures. The main dependent variables were selection time and incorrect mouse clicks missing the target item. In the post-experiment questionnaire the participants were asked to rank each of the menus on a 1-7 Likert scale according to the following four criteria: *Performance*, *Error rate*, *Frustration*, *Ease of use* (with 1 being the most negative and 7 being the most positive response to each question). The participants were also encouraged to provide free comments both through the questionnaire and informally.

### **4.3.3. Task**

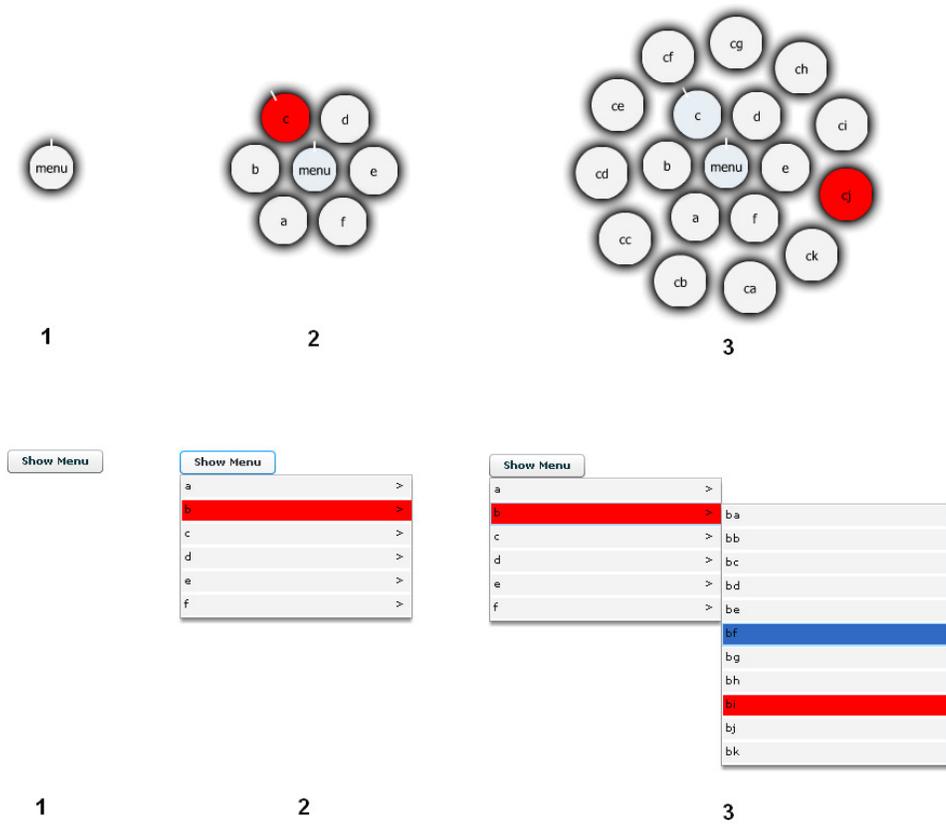
The task was to select a particular item from a menu hierarchy. All items in the selection path (i.e., parent items and the target item) were highlighted with the colour red. The whole path leading to the target item was clearly indicated (see Figure 4.2). Consequently, the participants did not have to search for an item nor decide which item to select next. As a result, the task emphasized navigation and minimized the contribution of other factors [Samp and Decker, 2011].

The above task is commonly used in menu research including studies modelling hierarchical menu selections with the compound ID [Ahlström, 2005] and comparisons of menu designs [Ahlström, 2005; Ahlström et. al., 2006; Cockburn and Gin, 2006; Tanvir et al., 2008].

Re-using the above task in the current experiment allows more direct comparison of the compound ID model and the SP model.

The menu content consisted of 6 items on the first level, 11 items on the second level, and 15 items on the third level (6-11-15). The intent of using a menu configuration of 6-11-15 was to fully populate the three rings (levels) in the CRL menu. This allowed for measuring the selection times across the whole circumference of the rings.

The menu labels consisted of one- to three-character labels to help the participants separate menu items visually; however, they did not have any meaning.



**Figure 4.2.** The task used in the experiment. Selection sequence for a target item on the second level for the CRL menu (at the top of the figure) and the CS menu (at the bottom of the figure). All items in a selection path are highlighted red thus the participants can focus on navigation and do not have to search or to decide which item to select next.

#### 4.3.4. Design

The experiment was based on a  $2 \times 3$  factorial design (*menu design*  $\times$  *target level*). The menu was either the CS menu or the CRL menu. The target item was on the first, second or third menu level.

One block of trials consisted of 45 menu selections. 10 tasks required the user to select an item from the first level, 15 tasks from the second level, and 20 tasks from the third level. Each participant completed a block of trials twice for each menu and before proceeding to the next menu. The item sequences for all the trials in a block were generated randomly, ensuring that all the menu items from all the levels were used at least once. This way we did not promote easy or difficult selection paths. Consequently,

the reported time for each level represents the average time it takes to select an item from a particular level for a given menu configuration. The order of the trials for each level was determined by a one-off random process, and reused in both menus across all the participants.

The random generation of item sequences prevented any learning effects.

#### ***4.3.5. Participants and Apparatus***

There were 28 participants, 8 women and 20 men, ranging from 20 to 28 years of age. All of them were experienced computer users using a mouse on a daily basis. Participants were rewarded financially for their time.

The experiment was conducted on an IBM T61p (core 2 duo 2.4GHz, 2GB of RAM) running Microsoft® Windows® XP.

#### ***4.3.6. Procedure***

The average session lasted approximately 40 minutes. The procedure was as follows.

- (1) Participants were told the procedure of the experiment (two menus, three levels, and two blocks). The task was described.
- (2) For both menus the system provided a two sentence description of their behaviour and allowed the participants to practice for two minutes.
- (3) The experiment started. The participants were asked to complete the selections as fast and as accurately as possible. Half of the participants started with the CRL menu and the other half with the CS menu. A block consisting of 45 menu selections started: 10 selections with a target item on the first level, 15 selections with a target item on the second level and 20 selections with a target item on the third level. There was a one minute break between the levels. The same block of 45 menu selections was repeated for the same menu after a two minute break. After the second block and a four minute break, participants proceeded to the second menu and repeated the same sequence of trials. Each trial started with the menu collapsed. After clicking on the central item (a rectangular button for the CS menu and a circle for the CRL menu) the first level menu was posted and the system started to measure the time. The task finished with the click on the target item. Time was stopped and the menu collapsed automatically. If a participant clicked outside the next highlighted item the trial was recorded as an error.

- (4) The participants completed the post-experiment questionnaire to rank the menus on qualitative dependent variables.

#### **4.4. Menu comparison**

This section compares the navigation performance of the CS and the CRL menu using the data collected in the experiment.

##### **4.4.1. Comparing menu designs**

The results from the second block were used to compare the task completion times. The menu configuration 6-11-15 might be regarded as unrealistic and could possibly bias the comparison outcomes. For this reason, the comparison also employs a subset of the results representing a smaller, more realistic menu configuration (6-8-8).

Two Two-way Repeated Measure (RM) ANOVAs (*menu type*  $\times$  *task level*) were run to determine significance of the completion time differences: one ANOVA for the 6-11-15 configuration and another for the 6-8-8 configuration. A separate ANOVA indicated that menu ordering and block did not have any significant effect.

##### **4.4.2. Meeting RM ANOVA assumptions**

Although RM ANOVA is a powerful statistical tool a set of assumptions have to be met to justify its use.

It can be observed in the Figure 4.3 that the standard deviations are widely different across the means. The ratio between the highest and the lowest standard deviation is 7:1 which is not acceptable. This is a common consequence of working with temporal data where the means increase together with the standard deviations. In such situations it is sensible to transform the measured times in order to equalize the variances and obtain times that more nearly approximate normal distribution within the conditional groups. To find a transformation we used an empirical technique based on regression of the log of standard deviations on the log of the means. If  $b$  is the estimated slope of the regression then the appropriate power for the transformation can be estimated as  $p = 1 - b$  [Maxwell and Delaney, 2004]. In our case  $b = 1.01$  thus  $p \sim 0$  which suggested using  $\log_{10}$  transformation. Note that in the context of measuring task completion time (i.e. latency) one could also consider reciprocal transformation, which would convert time to speed.

Although the transformation equalized variances across the groups the data was still non-spherical as the condition-difference variances were not homogenous. Mauchly's test indicated that the assumption of sphericity had been violated for *level*,  $\chi^2(2) = 10.91$ ,  $p < 0.01$ , and for *menu x level* interaction,  $\chi^2(2) = 6.77$ ,  $p < 0.05$ . It is important to note that within-subject designs are extremely sensitive to the violation of the sphericity assumption. A practical approach in such a situation is to use one of the corrections. In fact, many statistical books agree unanimously that unadjusted tests should never be used as they lead to unreliable conclusions (i.e. "highly significant" result cannot necessarily be trusted) [Maxwell and Delaney, 2004]. Therefore, the degrees of freedom were corrected using a conservative Greenhouse-Geisser correction ( $\epsilon = .74$ ,  $\epsilon = .81$  for *level* and *menu x level* respectively).

#### 4.4.3. Comparison of navigation performance

The task completion times are shown in Figure 4.3.

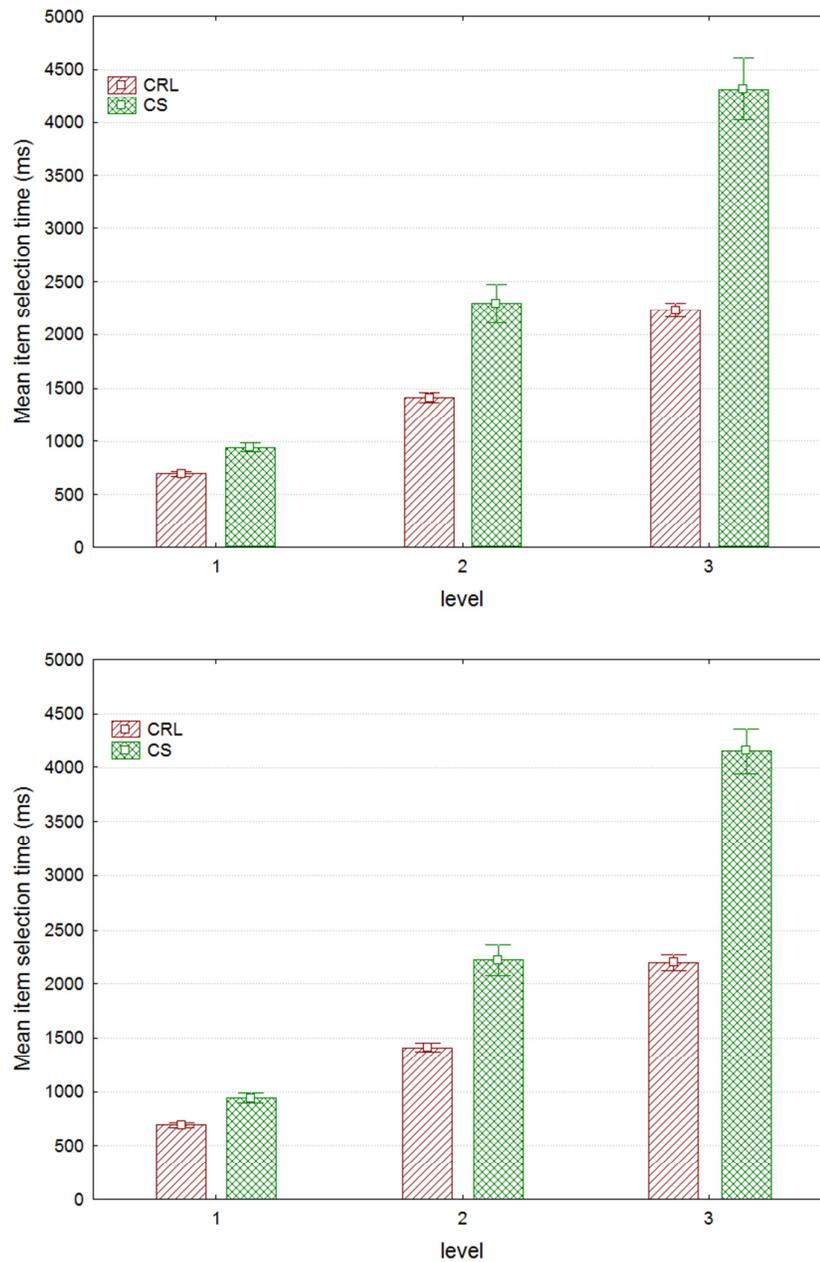
For both configurations there was significant main effect of *menu*, *level*, and *menu x level* interaction – all at level  $p < 0.01$  (see Table 4.1).

**Table 4.1. Results of ANOVAs (all effects sig. at level  $p < 0.01$ ).**

Menu configuration	Menu effect	Level effect	Menu x Level
6-11-15	$F_{1,27} = 317.1$	$F_{1.4, 38.7} = 9039$	$F_{1.6, 43} = 22.5$
6-8-8	$F_{1,27} = 259.4$	$F_{1.4, 37.8} = 7786$	$F_{1.7, 45.9} = 8.6$

The interaction effect is present because the times for the CS menu increase more rapidly with each level than for the CRL menu.

For post-hoc pair-wise comparisons we used a Bonferroni adjustment to preserve the family wise significance level. The CRL menu was faster than the CS menu on each level regardless of the menu configuration ( $p < 0.001$ ). For the 6-11-15 configuration the CRL menu was faster by 185 ms, 812 ms, and 2015 ms on the first, second, and third level respectively. For the 6-8-8 configuration the CRL menu was faster by 185 ms, 784 ms, and 1791 ms. A smaller configuration decreased the time differences between the menus but not considerably.

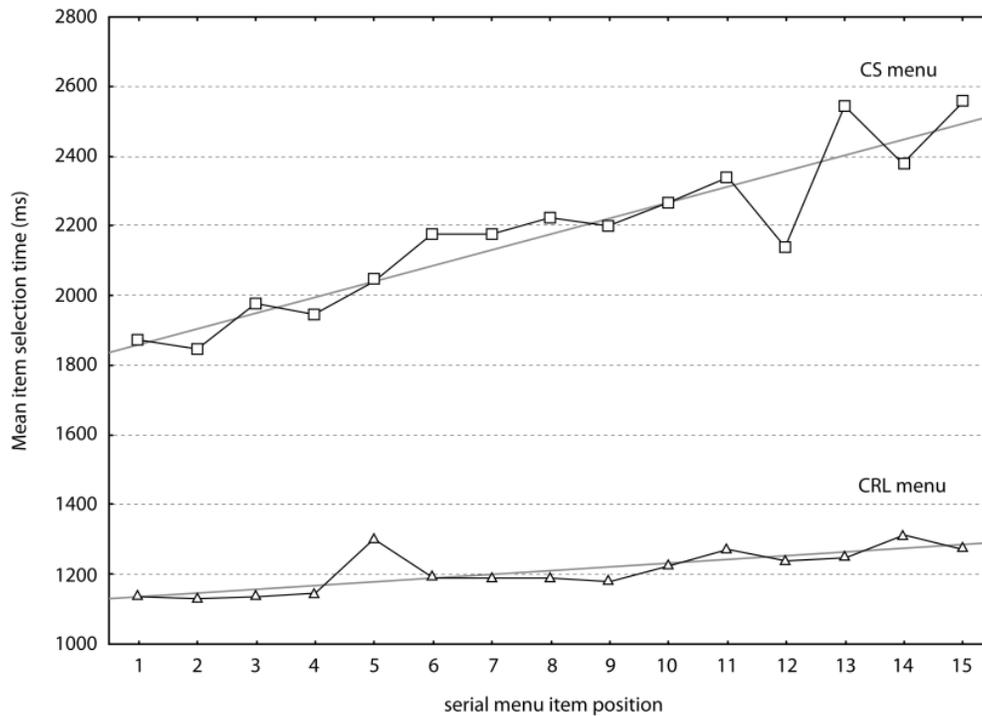


**Figure 4.3.** Mean task completion times for two menu designs as a function of level. On the top, the results are for the 6-11-15 configuration and on the bottom, the results are for the 6-8-8 configuration.

We also analysed the sub-selection times (i.e., time measured between selecting a parent item and one of its child items). For the CS menu the sub-selection times increase linearly with the item's serial position (see Figure 4.4). This supports prior

empirical evidence [Callahan et. al., 1988; Nilsen, 1991; Hornof and Kieras, 1997]. For the CRL menu we found a similar pattern: the selection time increases linearly with serial position (measured as a distance from a parent item). However, the slope of this increase is much flatter for the CRL menu and the sub-selection times for each level can be considered roughly constant. Similar findings were presented in [Callahan et. al., 1988] for single level radial menu. Extending this characteristic to hierarchical menus is an undoubtedly beneficial property of the CRL menu. All items on a given level are accessible within similar time frames.

1.6% of trials were erroneous (i.e., a mouse click missing a target item) for the CRL menu and 3.7% for the CS menu. However, the difference between the two menus was not significant.



**Figure 4.4.** Mean third sub-selection time for two menu designs as a function of an item's serial position. Note that the y axis (mean item selection times) starts with value 1000 ms, not 0 ms.

#### 4.4.4. Subjective perception

To analyse data from the questionnaire the Wilcoxon Signed-Rank test was used. The participants subjectively perceived the CRL menu as being faster, less error prone and

also less frustrating ( $p < .01$ ). However, no significance difference was found for ease of use.

24 participants said that they were faster with the CRL menu, 4 participants did not see any difference. All the participants claimed that they were more error prone with the CS menu. 18 participants were more frustrated with the CS menu, 4 with the CRL menu, and 6 did not see any difference. 6 participants considered the CRL menu easier to use, 5 the CS menu, and 17 considered both menus equal.

Subjective perception of performance supports the quantitative findings. With respect to errors, participants unanimously expressed that the CS menu was more error prone. This also resulted in increased frustration. However, the perception of errors is not supported by the number of erroneous mouse clicks. This is further investigated and discussed in [Samp and Decker, 2011].

## ***4.5. Modelling***

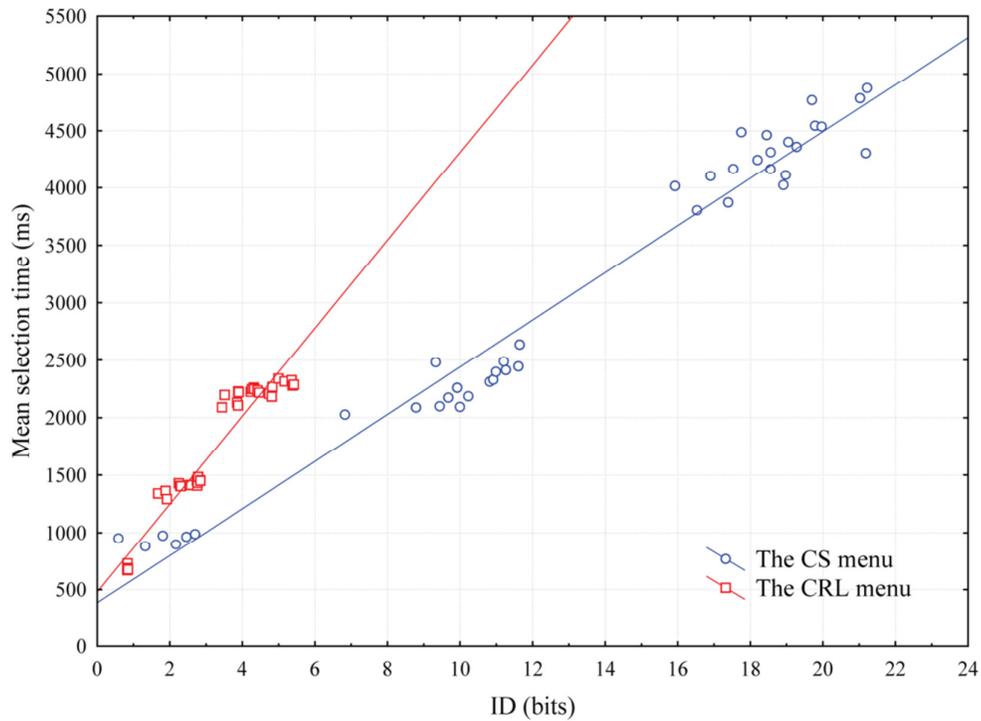
This section uses data collected in the experiment presented in Section 4.3 to build the model described in Section 4.2. Regression analysis is performed on the results from the second block for each menu after removing approximately 2.7% of the trials that were recorded as errors.

### ***4.5.1. Task difficulties***

The IDs of the 45 tasks ranged from 1 bit to 21.5 bits for the CS menu and from 1.18 bits to 6.06 bits for the CRL menu. The IDs for the CRL menu are more ‘squeezed’ as the result of decreased distances, increased sizes, and because the navigation consists only of Fitts’ tasks rather than both Fitts’ and steering tasks. The CRL menu decreased the theoretical difficulty of structurally equivalent tasks.

### ***4.5.2. The Sequential Pointing Model***

The modelling results are shown in Figure 4.5.



**Figure 4.5. Selection time as a function of task compound ID and menu design. For the CRL menu note low slope of selection times within each level and inter level increment suggesting substantial effects of sub-selection additive factors.**

Multiple regression produced strong models for the CS menu:

$$MT = 134.2 \times ID + 630.3 \times L + 475.4, R^2 = 0.973,$$

and for the CRL menu:

$$MT = 95.1 \times ID + 603.5 \times L + 592.5, R^2 = 0.996,$$

All parameters were significant at level  $p < 0.001$ .

The fit of the SP model is strong. In the following section we analyse more thoroughly model parameters.

#### **4.5.2.1. Model parameters**

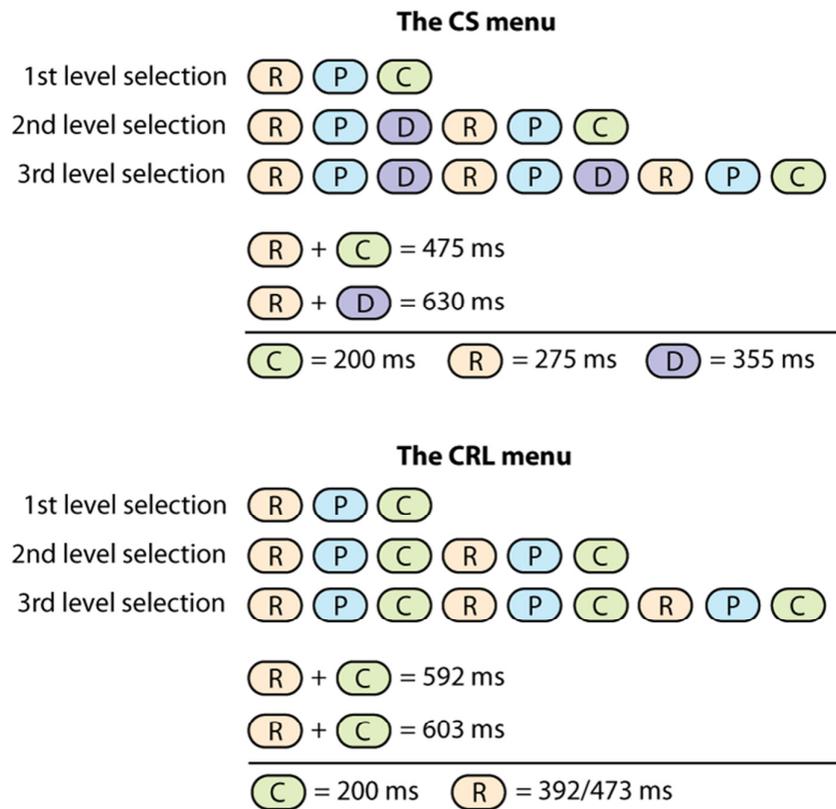
**Additive factors.** According to the model the additive factors associated with the final and preceding sub-selections (i.e., the parameters  $a$  and  $c$ ) are 475 ms and 630 ms respectively for the CS menu, and 603 ms and 592 ms for the CRL menu. The

model suggests that the additive factors for the two menus are different and that for the CS menu the additive factors of the final sub-selection differ from the additive factors of preceding sub-selections. To validate this interpretation in more depth we broke the task into sequences of operators of four types:

- R - Reaction time of aimed movement This includes reaction to visual stimuli and preparation of the corresponding movement, and it occurs for visually stimulated pointing tasks [Bekkering et. al., 1994] such as the one used in the experiment. For the current task the reaction time occurs each time a new sub-menu with the next highlighted item is posted.
- P - Pointing to the next item.
- C - Clicking a mouse button.
- D - Dwelling before a sub-menu is posted.

The operators represent menu design characteristics (i.e., C and D operators) as well as documented human psychomotor characteristics (i.e., P and R operators). Note that the R, D, and C operators are the additive components of the HCI pointing tasks identified in the previous works and mentioned earlier in Section 4.1.1.1.

Figure 4.6 shows sequences of the operators for the selections on three levels for both menus. For example, consider a selection from a second level using CS menu. After the menu is posted (note that the system started to measure time at this point, see Section 4.3.6) the user needs to react to visual stimuli and prepare a movement to a highlighted item (i.e., one R operator). Then, the user points at the item (i.e., one P operator). Once over the item, the user dwells for 1/3 sec (i.e., one D operator). After this short delay a second level sub-menu appears. The user needs to react to new visual stimuli and prepare a movement to the next highlighted item (i.e., one R operator). Then, the user points at the item in the second sub-menu (i.e., one P operator). And once over the item, the user clicks to finish the selection (i.e., one C operator). In summary, the selection on the second level using CS menu produces a sequence: R + P + D + R + P + C.



**Figure 4.6. Interaction sequences for each three-level menu. Operators used: R – reaction, P – pointing, C – click, D – delay. Values of the operators are computed using regression results.**

From Figure 4.6 it can be determined that for the CS menu the additive factors for the final sub-selection are R+C (i.e., one reaction time and one click) and for the preceding sub-selections R+D (i.e., one reaction time and one dwell). This is achieved by subtracting sequences of operators for the selections on any two subsequent levels. Following a similar process for the CRL menu, the additive factors for the final sub-selection are R+C and for the preceding sub-selections also R+C. The above sums can be compared to the corresponding model parameters (i.e., the parameter  $a$  and  $c$ ). For the CS menu one gets:

$$\begin{cases} R + C = 475 \text{ ms} \\ R + D = 630 \text{ ms} \end{cases} \quad (9)$$

and for the CRL menu:

$$\begin{cases} R + C = 592 \text{ ms} \\ R + C = 603 \text{ ms} \end{cases} \quad (10)$$

To resolve the equations we can use either  $C = 200 \text{ ms}$  [Card et. al., 1983, p. 264] or  $D = 1/3 \text{ sec}$ . Both approaches lead to similar results. Using  $C = 200 \text{ ms}$  the values of the operators are as follows. For the CS menu:

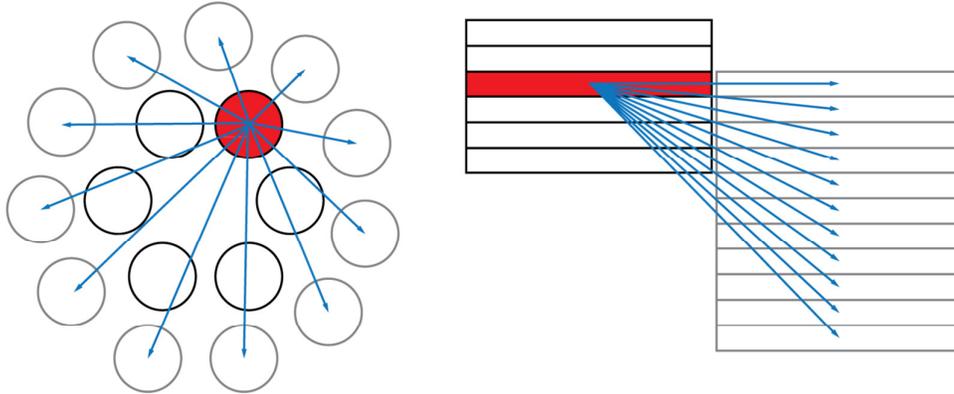
$$\begin{aligned} C &= 200 \text{ ms} \\ R &= 275 \text{ ms} \\ D &= 355 \text{ ms} \end{aligned}$$

and for the CRL menu:

$$\begin{aligned} C &= 200 \text{ ms} \\ R &= 392 \text{ ms or } R = 403 \text{ ms} \end{aligned}$$

The above analysis supports the validity of the SP model in four ways. First, for the CRL menu the final and preceding sub-selections involve the same additive factors (i.e.,  $R+C$ ). The model parameters suggest the same; that is, the parameters  $a$  and  $c$  which correspond to the additive factors of the final and preceding sub-selections respectively are practically the same (i.e., 592 ms and 603 ms). Second, for the CS menu the final and preceding sub-selections involve different additive factors (i.e.,  $R+C$  and  $R+D$ ). The model suggests the same; that is, the parameters  $a$  and  $c$  are considerably different. Moreover, the value of 475 ms corresponding to  $R+C$  is smaller than the value of 630 ms corresponding to  $R+D$  which is supported by the fact that  $C = 200 \text{ ms}$  [Card et. al., 1983, p. 264] is smaller than the implemented  $D = 1/3 \text{ sec}$ . Third, the calculated values of the  $D$  operator are 355 ms which closely match the implemented value of  $1/3 \text{ sec}$ . Fourth, the values of the  $R$  operator match previous findings regarding reaction time [Card et. al., 1983, p. 66 define simple reaction time range as 100-400 ms based on extensive prior research]. In particular, the 275 ms for the CS menu is almost identical to  $234 \pm 41 \text{ ms}$ , which has been reported as the reaction time of aimed movement to visual stimuli [Bekkering et. al., 1994]. The  $R$  operator for the CRL menu is approximately 122 ms bigger than for the CS menu. We

believe that this difference is expected due to the wider spatial extent of the CRL sub-menus compared to the CS sub-menus (see Figure 4.7). Prior work shows that the reaction time of aimed movements is increased when the spatial extent of the area where the next target item may appear increases [Bock and Eversheim, 2000].



**Figure 4.7.** For the CRL menu (on the left) the next target item can appear in any direction from the parent item, thus the spatial extent is large. For the CS menu (on the right) the next target item can only appear to the bottom right of the parent item, thus the spatial extent is smaller. The larger spatial extent of the CRL menu leads to larger reaction time before the next movement commences.

It is important to note that the reaction time (R operator) in the task represents the reaction time of a pointing movement to a highlighted item. In real scenarios users will not see the item they need to select highlighted. Thus the R operator is not part of the novice or expert navigation performance but rather is specific to the experimental task.

The analysis based on sequences of operators assumes that the actions represented by the operators occur in succession. However, a concern remains that dwelling and mouse movements may occur in parallel rather than in succession. A similar concern exists for visual search and pointing in menus. That is, visual search and pointing may occur in parallel. However, both are successfully modeled as serial events not only by predictive models which are typically within 2% of empirical data [Cockburn et. al., 2007; Ahlström, 2005; Ahlström et. al., 2010] but also by models explaining detailed user behaviour [Hornof and Kieras, 1997; Anderson et. al., 1997; Anderson et. al., 1998; Byrne, 2001]. Further, the KLM model adds sequences of operators, which theoretically may occur in parallel, and yields accurate predictions in a variety of

scenarios. The extent to which the results presented above match both implemented menu components and previously studied components from existing HCI/Cognitive Psychology research indicates that, as in the case of the other models, the parallelism, if any, has little practical effect.

**Slope.** According to the model, the motor processing rate or information aspect of the input is 95 ms/bit for the CRL menu and 134 ms/bit for the CS menu. The slopes of the SP model fall into the range established by extensive prior work on modelling pointing tasks, and in particular work on modelling selections in real menu designs (see Section 4.1.1.1). The CS menu has a larger slope than the CRL menu. This indicates that participants were biased more towards accuracy for the CS menu than for the CRL menu.

The following section discusses explanations provided by the SP model as to how the tested menus differ in terms of navigation.

## **4.6. Discussion**

The IDs for the CS menu ranged from 1 bit to 21.5 bits while for the CRL menu from 1.18 bits to 6.06 bits. The CRL menu decreased the difficulty (ID) of structurally equivalent tasks – that is, the CRL menu required fewer bits of information to be transmitted through the human motor channel to reach the same outcome.

The additive factors accounted for approximately 552 ms (45%) and 597 ms (80%) of the total selection time for the CS menu and the CRL menu respectively. For the CRL menu the additive factors associated with the final and the preceding sub-selections are the same while for the CS menu they are different and bigger for the preceding sub-selections.

The different slopes for the two menus indicated that participants used a different strategic integration of perceptual and motor processing, or microstrategy [Gray and Boehm-Davis, 2000]. In other words, each menu biased participants towards different speed/accuracy trade-off characteristics.

People are sensitive to small changes in artifact design, are aware of their own limitations and tend to select a microstrategy which is the most efficient, in terms of performance but also error rate, in the particular task context [Augustyn and Rosenbaum, 2005; Gray and Boehm-Davis, 2000].

The CS menu required considerably more accuracy than the CRL menu while navigating (i.e., 39 ms (41%) more per bit). This need for accuracy can be explained with smaller item sizes, longer distances, inclusion of steering, and increased chances of committing a navigation error in the CS menu (refs here from manuscript).

The CRL menu required less accuracy than the CS menu and allowed for more prompt selections. The CRL menu reaches the limits of human motor performance with a mouse pointing device. This indicates that the design does not hamper navigation in any notable way. It can be asserted that the design parameters for the CRL menu, such as item sizes and spacing between items and levels, do not force users to increase accuracy and thus do not need adjustments.

In terms of menu navigation, the CRL menu gains an edge over the CS menu by lowering the difficulty of the tasks and by inducing users to more prompt and less accurate movements. This supports the beneficial design of the CRL menu, particularly for expert users who spend most of the selection time navigating to the appropriate items.

#### **4.7. Conclusions**

This chapter presented a study that focused on navigation times in two menus: a linear cascading menu and the CRL menu. The study employed qualitative and quantitative measures. The results were used to compare the tested menus and to build a model of menu navigation.

Compared to the CS menu, the CRL menu decreased the difficulty of structurally equivalent selection tasks by approximately 300%. The CRL menu enabled participants to operate close to the limits of the human motor processing rate. The CS menu, on the other hand, induced participants to bias their strategy of perceptual and motor integration towards accuracy; thus sacrificing speed and preventing more unprompted navigation. Design parameters of the CS menu hamper navigation and require users to invest more effort when navigating.

The comparison of the selection times supports the conclusions provided by the SP model. The CRL menu yielded approximately 34% better navigation performance. For the CRL menu, all items on the same menu level are accessible roughly within the same time frame. The subjective perception also spoke in favour of the CRL menu.



## ***5. Navigation Time Variability: Measuring Menu Navigation Errors***

There has been a great deal of research into alternative menu designs, including numerous cascading menu improvements [Kobayashi and Igarashi, 2003; Ahlström, 2005; Cockburn and Gin, 2006; Tanvir et. al., 2008], radial menus [Samp and Decker, 2010], marking menus [Kurtenbach, 1993], and much more.

Research studies of menu designs typically focus on measuring performance in the form of selection times and error rates in the form of the number of clicks missing the target menu item. Sometimes the subjective perception is also measured with a post-experiment questionnaire.

When it comes to errors, it turns out that the error rates are low, typically below 5% (e.g., [Ellis et. al., 1995; Ahlström, 2005; Cockburn and Gin, 2006; Tanvir et. al., 2008; Samp and Decker, 2010]). The error rates are often not significantly different for tested designs. Some studies abandon the analysis of errors, concluding that the errors are too sparse to provide any interesting insights (e.g., [Cockburn and Gin, 2006]). These results would suggest that errors do not play an important role in menu selection.

It is important to note, however, that the traditional way of measuring errors focuses only on one particular type of navigation error—i.e., clicks missing a target item. However, navigation does not consist only of mouse clicks but of all motor actions that the user needs to perform when selecting from a menu (e.g., dwelling or moving the mouse pointer). These other actions can result in errors which are not captured by counting the number of incorrect mouse clicks. For example, in the cascading menu small steering errors causing incorrect selection changes or unexpected sub-menu disappearance do not increase number of incorrect mouse clicks. Navigation is an important part of menu selection, concerning both novices and experts (see Section 3.2). Therefore, it is important to understand the problem of menu navigation errors more thoroughly.

This chapter investigates menu navigation errors in more depth. We propose the Navigation Time Variability (NTV) measure to address the aforementioned problems. It captures the total severity of navigation errors. The severity is understood as the time needed to recover from the errors committed. The data required by the measure is typically collected in menu studies. We describe the measure and use it on data collected in the experiment presented in Section 4.3. The results demonstrate that the CRL menu results in considerably less navigation errors than the CS menu.

The rest of this chapter is structured as follows. Section 5.1 presents the related research. Section 5.2 describes the measure. Section 5.3 presents the use of the measure on the data collected in the experiment presented in Section 4.3. The results are discussed. Section 5.4 finishes the chapter with the conclusions.

### **5.1. Related Research**

As discussed in Section 3.2, there are three components of menu selection: visual search, decision, and navigation [Cockburn et al., 2007]. Novices search for an item and then navigate to it. Experts, on the other hand, do not have to search for menu items as they know their locations. They decide which item to select and then navigate to it. If the menu is hierarchical, search and navigation (novices) or decision and navigation (experts) are performed multiple times for each menu level.

Navigation concerns novices and experts. It refers to all motor actions that the user needs to perform when selecting from a menu (e.g., moving the mouse pointer, dwelling, mouse clicks). Navigation is an important part of menu selection. Novices and experts can spend between 20-50% of the total selection time on navigation [Samp and Decker, 2010].

There are many possible differences in how menus are navigated. Some menus employ a point-and-click interaction style [Samp and Decker, 2010], others a dragging interaction style [Kurtenbach, 1993]. Some menus restrict navigation trajectory when moving between the sub-menus, others do not [Cockburn and Gin, 2006; Tanvir et al., 2008]. Some menus make the navigation to frequently selected items easier by dynamically increasing their sizes [Cockburn et al., 2007]. Some menus facilitate navigation between the sub-menus; for example, by attracting the cursor towards an open sub-menu [Ahlström, 2005] or by opening a sub-menu faster if the cursor moves towards it [Kobayashi and Igarashi, 2003]. These are only a few examples. The plethora

of menu navigation techniques makes it important to understand how error prone they are, beyond the number of clicks missing target items.

The importance of navigation errors is informally established in the HCI community. Numerous cascading menu improvements are motivated by various navigation problems [Kobayashi and Igarashi, 2003; Ahlström, 2005; Cockburn and Gin, 2006; Tanvir et. al., 2008]. Pastel [Pastel, 2006] shows that steering through sharp corners, like in the case of the cascading menu, can induce errors. Kobayashi [Kobayashi and Igarashi, 2003] provides empirical evidence that the number of unexpected submenu appearances, again for the cascading menu, can be substantial. These works hint at some navigation problems beyond traditional error rate but are quite specific in what they focus on (i.e., one type of error particular for the cascading menu). None of the work has focused on a general approach to measuring the total impact of all navigation errors.

To address this problem, an approach focusing on measuring errors of pointing devices could be considered. MacKenzie et al. [MacKenzie et. al., 2001] proposed seven accuracy measures to elicit differences among pointing devices in precision pointing tasks :1) target re-entry 2) axis crossing, 3) movement direction change, 4) orthogonal direction change, 5) movement variability, 6) movement error, and 7) movement offset. The measures, however, do not necessarily represent menu navigation errors. Axis-crossing or movement changes do not have to result in errors such as sub-menu disappearance or selection changes – i.e., errors that the user needs to correct. This will depend on particular menu design and size of the committed error. The above measures are well suited to assess deviations from an optimal pointing solution. It is arguable, however, if such a general optimal pointing solution exists for hierarchical menus (e.g., should the steering finish at the border of a sub-menu, in its center, or somewhere else?).

The above approach and other approaches based on counting specific types of errors also pose demands on the experimental software. All the possible errors have to be tracked individually. This might be difficult in some environments such as those employing third-party applications or toolkits.

We aim at creating a simpler and more abstract approach. We want to refrain from listing a priori all possible types of navigation errors and tracking these in the experimental software. Our goal is to assess the total impact of all navigation errors,

not contribution of the individual predetermined types of errors. Such an approach would enable more immediate view of the problem of navigation errors in menus.

## ***5.2. Navigation Time Variability measure***

There are many possible sources of navigation errors stemming from the many different ways of navigating menus (some examples mentioned in Section 5.1).

We propose the following view of the problem of navigation errors: A user trying to be as fast and as accurate as possible should be able to navigate to the same target item multiple times within a similar time frame. An increased variability in navigation time for the same target item indicates navigation errors. This is because recovering from navigation errors (e.g., re-pointing to a target item) requires additional time which is not strictly related to navigating towards a desired item.

Note that the increased variability is not necessarily connected with an event such as sub-menu disappearance. The corrections done by the users to prevent the errors—for example, a temporary change of speed-accuracy strategy preventing sub-menu disappearance—will also increase variability. This is a desired behaviour because such corrections also indicate navigation difficulties which are the object of our interest.

According to the above view, what we focus on is not the occurrence of a particular event indicating an error but rather the occurrence of variability in navigation time indicating extra time spent on recovering from errors. Consequently, it is not we who decide where the navigation error occurred, but rather the user by making necessary, time-consuming corrections.

The above view of navigation errors takes into account severity of the errors. Severity is an important aspect of errors having a strong effect on user perception [Feng and Sears, 2010]. In our case, the more severe navigation errors require more time to recover and thus lead to larger variability. This is important because we can expect different types of navigation errors to cause different degree of difficulties.

To formalize the described variability we propose the *Navigation Time Variability* measure (NTV). It is calculated as follows. (1) For each participant we establish the min-max range of navigation times obtained for **the same menu item** with the same menu design. We did not use standard deviation instead of min-max as we expect small number of measurements per item per participant (i.e., two or three—we discuss

this further in Section 5.2.1). Min-max range also assures easy interpretation of the results as it represents the difference between the fastest and the slowest navigation to the same menu item. (2) For each participant, we calculate the average of the min-max ranges across the different menu items. The result is one value per participant representing a single NTV score.

Since the NTV is computed on per participant basis, the inferential statistics can be used to seek significant differences between tested menus.

The NTV measure has four important characteristics. First, the sources of navigation errors do not have to be known in advance. Second, it focuses on all the errors which truly require user time and effort to be corrected. Third, the measure is based on severity of the errors—i.e., more severe navigation errors result in higher scores. Fourth, the measure assures meaningful interpretation—i.e., it expresses, in a statistical sense, how much additional time is required to correct the committed errors. In light of these characteristics, it becomes clear that the measure is not suited to identify sources of the errors or the contribution of the predetermined types of errors. Rather, its goal is to assess and compare the total impact of the navigation errors.

### ***5.2.1. Measuring navigation times – practical considerations***

The navigation times have to be collected in an experiment. However, the experimental task cannot consist only of navigation. Consequently, some variability in collected times (i.e., menu selection times) might be also attributed to other components present in the experimental task. For example, if the task requires a simple decision apart from navigation, the variability of the decision time will contribute to the total measured variability. Therefore, if the goal is to assess navigation errors, it is important to use a task that emphasizes navigation and minimizes contribution of the other components (i.e., the other components should be relatively small compared to navigation and have small variability ranges). In particular, the menu selection task should not require visual search or problem solving because both can take long and introduce extensive variability.

If the task emphasizes navigation, then the NTV computed on selection times will allow one to assess qualitatively which menu causes fewer navigation errors. However, if the goal is to assess the exact quantitative extent of the navigation errors, the contribution of the other components have to be factored out. To this end, the components have to be known and have known variability ranges.

A considerable amount of menu studies have focused on measuring navigation performance. They employ a common task which emphasizes navigation and adheres to the above characteristics. The task is to select an item from a single or hierarchical menu. All items in a selection path (i.e., parent items and the target item) are highlighted. Consequently, the participants do not have to search for each item nor decide which item to select. The participants only need to: 1) respond to visual stimuli and prepare the movement which takes  $234 \pm 41$  ms [Bekkering et. al., 1994] and then 2) navigate to the item. Ahlstrom [Ahlström, 2005] demonstrated empirically that the above task emphasizes navigation. He accurately modeled total selection times using only navigation component based on the Fitts' and the steering pointing laws.

The above task is commonly used in menu studies (e.g., [Ahlström, 2005; Cockburn and Gin, 2006; Tanvir et. al., 2008]). It is also common to administer two or more blocks of menu selections for each tested menu design (e.g., [Ahlström, 2005; Cockburn et. al., 2007]). Therefore, there is potential for similar menu studies to use the proposed measure without modifying the experimental design but merely by extending the error analysis part.

### ***5.3. Experiment***

This section compares navigation problems of the CS and the CRL menu using the NTV measure. The necessary data was collected in the experiment presented in Section 4.3. This time, however, the results from both blocks are used.

#### ***5.3.1. Results and Discussion***

The traditional measure of error rate indicated that 1.6% of the trials were erroneous (i.e., a mouse click missing a target item) for the CRL menu and 3.7% for the CS menu. Because the scores of error rates were sparse and not normally distributed we used the Wilcoxon Signed-Rank test for statistical analysis. The test indicated that the error rate is not significantly different for both menus ( $p > 0.05$ ).

The results of the questionnaire were as follows. All the participants claimed that they were more error prone with the CS menu. 18 participants were more frustrated with the CS menu, 4 with the CRL menu, and 6 did not see any difference. 6 participants considered the CRL menu easier to use, 5 the CS menu, and 17 considered both menus equal. Because the subjective scores are non-parametric, we again used the Wilcoxon Signed-Rank test. The CRL menu is perceived as significantly less

erroneous ( $p < 0.01$ ) and less frustrating ( $p < 0.01$ ) than the CS menu. No difference was found with respect to ease of use.

The results indicate that the CS menu is strongly perceived as more erroneous and more frustrating. However, this perception cannot be attributed solely to the error rate (i.e., the number of clicks missing target items) as it was not significantly different for both menus. The subjective perception is not supported quantitatively. The results hint that there is more to navigation errors than clicks missing their target items.

Using data from two blocks, we calculated the NTV for each participant. Table 5.1 shows the summary of the results.

**Table 5.1. The NTV on the three menu levels averaged across the participants. The NTV scores followed normal distribution within each menu x level cell.**

	Level 1	Level 2	Level 3	Marginal mean
CRL MENU	85 ms	164 ms	232 ms	160 ms
CS MENU	176 ms	862 ms	1005 ms	681 ms

To determine if the NTV differences are significant, we performed pair-wise comparisons using dependent measures t-test. The family-wise significance level was adjusted according to the number of tests performed. The comparisons revealed that the menus were not different on the first level but the CRL menu generated lower NTV than the CS menu on the second and the third level. For the CS menu, each level produced higher NTV compared to the previous levels. For the CRL menu, the first level had lower NTV than the second and third levels which did not differ between themselves. All the reported differences are at level  $p < 0.001$ .

**The CRL menu.** For the CRL menu, the NTV on the first level is the lowest (85 ms). It can be attributed to the variability of reaction time of aimed movements (82 ms) [Bekkering et. al., 1994]). The NTV on the second and the third level are doubled and tripled respectively as these levels require one additional reaction time compared to the previous level. There is no sign of any additional navigation variability which could indicate navigation errors. The participants were able to maintain roughly constant navigation performance on each level when selecting the same items.

**The CS menu.** For the CS menu, the NTV rapidly decreases between the levels. This indicates that participants, who performed well with the CRL menu, had more problems with the CS menu. The size of the NTV for the CS menu and the size of the differences in the NTV between both menus certainly cannot be attributed to the variability of reaction time of aimed movements. Furthermore, as the NTV for both menus is similar for the first level, we conclude that it is the navigation between the levels that causes rapid increase of the NTV for the CS menu. After factoring out the variability of reaction time of aimed movements from the NTV scores in Table 5.1, we estimate that recovering from the navigation errors takes approximately 94 ms, 698 ms, and 759 ms for selections on the first, the second, and the third level respectively. These amounts are substantial compared to the typical total selection times—e.g., according to [Tanvir et. al., 2008], approximately 1750 ms and 2500 ms for the second and the third level selections. This hints at the importance of navigation errors.

The above results provide quantitative support for the subjective findings regarding errors. To remind the reader, the CS menu was perceived as significantly more erroneous (stated unanimously) and significantly more frustrating than the CRL menu. In contrast to the traditional error rate, the NTV captured this difference.

Finally, the results also demonstrate that the NTV can be small, even for selections on the third level. This finding is important because it supports the underlying assumption of the NTV measure stating that a user can navigate to the same menu item within a similar time frame.

## **5.4. Conclusions**

This chapter proposed a new measure of navigation errors in menus and used it to compare the CS and the CRL menu.

We demonstrated that the proposed NTV measure gives quantitative information on navigation errors beyond traditional measure of error rate. The measure is not intended to replace the traditional measures (e.g., error rates). Rather, we consider it supplementary measure, with the potential to assess the total impact of navigation errors.

The results showed that the CRL menu results in considerably less navigation errors than the CS menu. These results were supported with the subjective responses.

Less navigation errors decrease required precision and enable more unprompted selections which are characteristic for experts. Moreover, less precision might also increase the upper limit of skilled performance.



## ***6. Supporting Menu Design with Radial Layouts***

The two previous chapters explored navigation performance of the CS and the CRL menu. This chapter focuses on layouts. A study compares two main characteristics of radial and linear layouts: a) the time it takes to find an item (i.e. visual search time), b) the time it takes to navigate to an item (i.e. pointing time). Objective and subjective measures are used, two menu sizes, three menu levels, one linear and three radial layout variations. The three radial layout variations are those presented in Section 3.4.2. The collected data is also used to assess novice and expert performance of all tested designs. Finally, the study assesses also screen consumption of the tested designs.

The rest of the chapter is structured as follows. Section 6.1 presents an experiment which investigates the demands posed by the radial layout and the linear layout on visual search and pointing. Section 6.2 reports the results, and Section 6.3 discusses the findings. Section 6.4 finishes the chapter with the conclusions.

### ***6.1. Experiment***

Previous work focused mainly on “design versus design” studies in novice or expert scenarios. The objective of this experiment is to investigate how radial layout compares to linear layout in terms of visual search and pointing performance. The experiment includes two menu sizes, three levels, one linear and three radial menu variations. We are interested in objective and subjective measures. Finally, the collected data will serve to assess novice and expert performance of tested designs.

In order to measure the time required for visual search, we need to distinguish between visual search and pointing times.

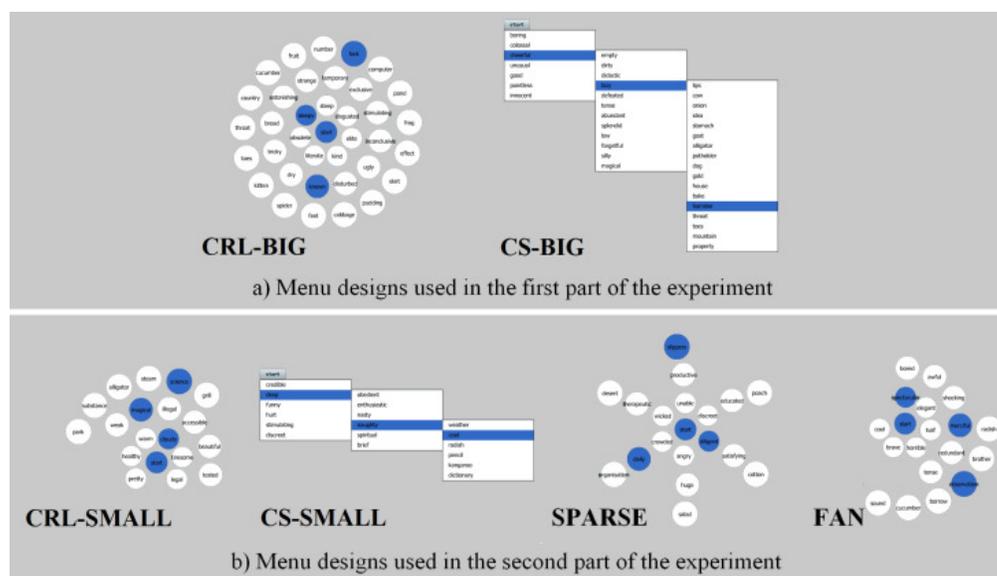


Figure 6.1. Menu designs used in the experiment.

### 6.1.1. Separating Visual Search from Pointing

In typical menu selection experiments, visual search and mouse movements times are measured as a single value. For this reason it is not clear how much of the total time should be attributed to either visual search or pointing. To separate both, we adopted the Point-Completion Deadline method proposed in [Hornof, 2001]:

*“Once the mouse starts to move, the participant is given a limited amount of time to click on the target; if the participant does not click the mouse button before this point-completion deadline is reached, then the trial is interrupted and recorded as an error. Thus, participants are discouraged from moving the mouse until they have found the target, and the start of the mouse movement consistently marks the end of visual search.”*

For our experiment, the point-completion deadline times were computed using Welford’s version of Fitts’ law with constant  $a$  and  $b$  set to 300 as suggested in [Hornof, 2001]. The item widths required by Fitts’ law were measured along the line of approach as suggested in [MacKenzie, 1992a].

### 6.1.2. Menus

The CRL menu, its two variations and one linear menu were used in the experiment. The cascading pop-up menu (CS) was chosen to represent typical linear context menus. The implemented menu is equivalent to those found in contemporary

applications, such as MS Word, with two exceptions: 1) an explicit click on each level is necessary to open the submenu (i.e. submenus do not open after dwelling on a parent item); 2) diagonal movements from a parent item toward its submenu are possible without a risk of the submenu closing. These two exceptions eliminated an additional factor (click or dwell) from the experiment, kept the interaction technique for linear and radial menus the same, and allowed us to focus on the layout differences. Furthermore, while the search times will be the same, the pointing times for our CS menu will be lower than for other CS variants using dwelling and/or restricting diagonal movements. This is because mouse clicks are faster than typical dwelling times (~200 ms [Card et. al., 1983] versus 333 ms) and diagonal movements are faster than the combination of horizontal and vertical movements [Ahlström, 2005]. Therefore, we can safely assume that our CS design yields faster selections than other typical CS designs which do not have the mentioned exceptions. Consequently, we can use measured times as a lower bound for typical CS designs.

To determine the size of the items in the CS menu, we analysed several menus of widely used applications and found that widths varied significantly. We chose the size of 190×21 pixels (area=3990 pixels) for the CS menu because this size could accommodate three word labels. To determine the size of the items for the radial menus, the volume of our circular items was not chosen to match the volume of our representative cascading menu item. Rather, we experimentally determined the size for the circles such that they provided enough space to also accommodate three word labels (eventually broken to two or three lines) while avoiding overlaps (see Section 3.4.3). The diameters for the circular items were 44 (a=1134p), 50 (a=1521p), and 54 (a=1810p) pixels for the first, second, and third level respectively.

### **6.1.3. Task**

The menu content was constructed in the following way: the first and second levels contained adjectives from a dictionary of 600 commonly used adjectives; the third level contained nouns from a dictionary of 400 commonly used nouns.

The task was to select a three word phrase of the form adjective-adjective-noun, such as “cloudy magical science”, from a three-level menu. The target phrases were constructed this way to make them easy to remember. Arbitrary three word phrases may be hard to memorize and recall during the course of the trial, and may negatively affect the measured times. Additionally, our approach allowed us to easily create

content with an arbitrary menu size and desired degree of randomness. To prevent any learning effects and force participants to visually search for the items, a new random target phrase and menu content was generated for each trial and for each participant independently. Menu items were randomly distributed within menu levels. Consequently, our task simulated novice behaviour.

First, the target phrase was shown on a blank page. After reading and memorizing it, the participant proceeded with a mouse click. At this point the phrase disappeared and the menu was displayed. The target phrase disappeared in order to prevent rescans which would bias the visual search results [Aaltonen et. al., 1998]. The participant performed a task consisting of three selections, one word on each level.

The experimental setup imposed a point-completion deadline. In contrast to the study presented in [Hornof, 2001], our experiment exploits this technique in a task comprising a series of selections. On each level, once the participant moved a mouse pointer by more than 5 pixels, he or she had a limited time for completing the next selection. If the participant did not select the next menu item before the deadline, the trial was stopped, a buzzer sound played, an alert message appeared reminding the participant to find an item before moving the mouse, and the trial was recorded as an error. After selecting an appropriate item, a new level appeared and the participant had to again find the next item before starting to move the mouse.

### **6.1.4. Design**

#### **6.1.4.1. Part 1 – Big Menus**

The first part of the experiment included one radial and one linear menu, both containing 7, 12, and 17 items on the first, second, and third level respectively (1428 items in total). This configuration fully populated all three rings in the radial menu creating a situation where participants had to possibly search the entire circumference of the ring to find a menu item (see Figure 6.1a).

The sequences for the target phrases were randomly generated for each participant, assuring that the physical selection path lengths were evenly distributed within the range of all possible lengths. This way we did not promote easy or complex tasks on per participant basis. Each participant completed two blocks of 40 trials, one block for each menu. Half of the participants started with the radial menu and the other half with the linear menu. Tasks were presented to each participant in a random order.

#### **6.1.4.2. Part 2 –Small Menus**

In the second part of the experiment, we used a smaller, more realistic menu configuration containing six items on all three levels (216 items in total). We used one linear and three previously introduced radial menu variations (see Figure 6.1b, and Section 3.4.2) to test if the menu size and different techniques of laying out the items around the rings affect the performance.

Each participant completed four blocks of 18 trials, one block for each menu. For each menu type, we added two extra warm-up tasks which were excluded from the analysis. We used a Latin square to control the menu type order. Tasks were presented to each participant in a random order.

#### **6.1.5. Measures**

The experiment measured the following quantitative dependent variables: a) visual search time – measured between a mouse click and the following mouse move; b) pointing time – measured between beginning of a mouse move and the following mouse click; c) number of errors resulting from mouse clicks missing the targets. For each task, the visual search and pointing times were recorded for all three selections, one for each level.

For the qualitative analysis we administered two questionnaires measuring subjective attitude of the participants, and a free form interview. The first questionnaire was administered after the first part of the experiment. Participants were reminded that the task consisted of: a) finding an item and then b) pointing and selecting it. Then, they had to choose a menu for which **finding** an item was **easier** (first self-reported variable) (linear, radial, no difference) and a menu for which finding an item was **faster** (second variable). Similarly, participants had to choose a menu for which **selecting** was **easier** (third variable) and **faster** (fourth variable). A textual explanation had to be provided for all the answers.

The second questionnaire was administered after the second part of the experiment. The participants had to rank four menus on a 1-5 Likert scale according to two questions: “*How easy was it to **find** items using the menu X?*” and “*How easy was it to **select** items using the menu X?*”. Further, the participants had to order four menus according to their preference for the tasks performed. The questionnaires were anonymous.

### **6.1.6. Participants and apparatus**

A total of 18 participants (5 female and 13 male) were recruited for the experiment. None of them had used the CRL menu before. All were between the ages of 22-30 and used a computer and a mouse on daily basis. All participants received sweets.

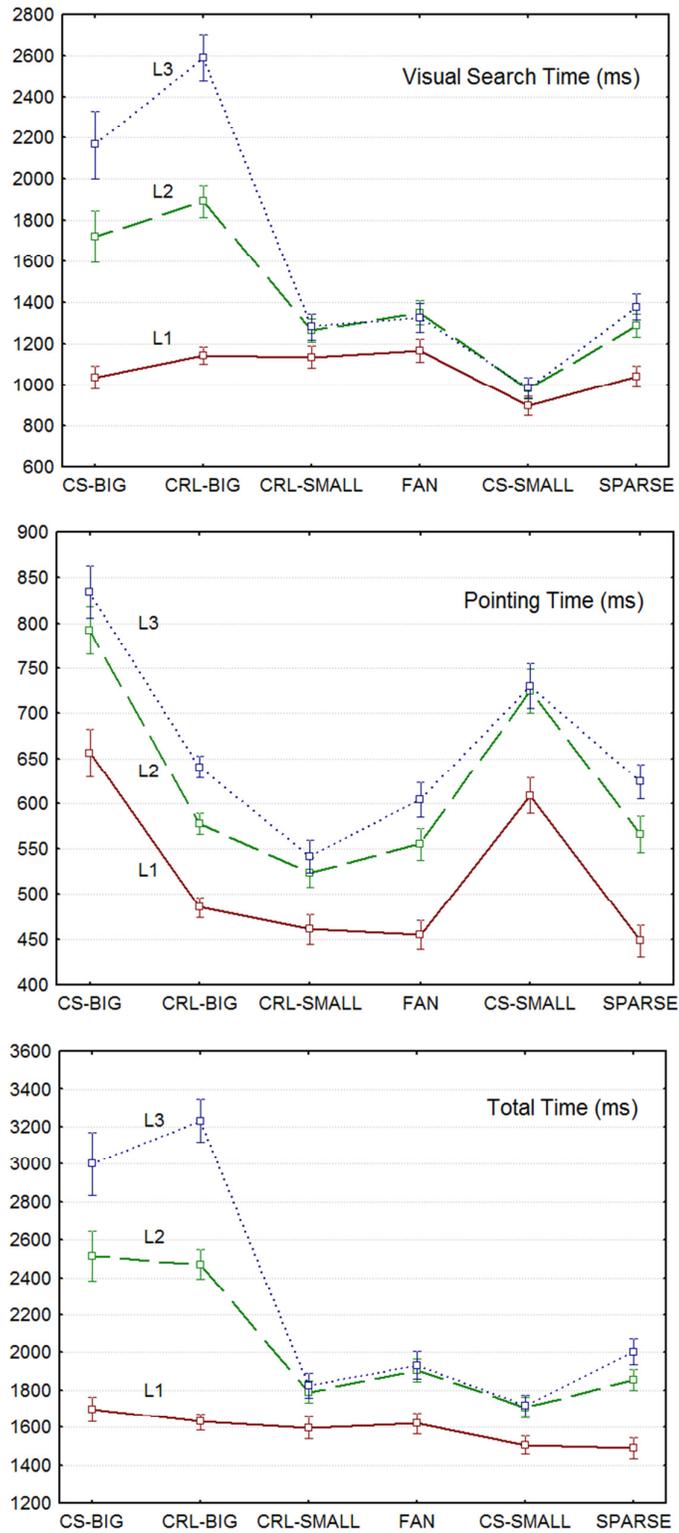
The experiment was conducted on an IBM T61p laptop running MS Windows XP. The connected monitor was a 20" IBM ThinkVision L200p LCD and used its native resolution of 1600x1200. A mouse was used to perform the tasks. The whole environment resembled typical office conditions.

### **6.1.7. Procedure**

The average session lasted approximately 40 minutes per participant. The procedure was as follows. 1) The participants were briefly introduced to the experiment and explained that they should select an item only after finding it visually. 2) In a practice session participants became comfortable with the task. 3) The first part of the experiment started followed by the first questionnaire. After a short break they proceeded to the second part of the experiment followed by the second questionnaire. 4) The participants were interviewed. They were encouraged to freely express their comments, observations, and suggestions.

## **6.2. Results**

The results are graphically presented in Figure 6.2. For clarity, standard deviations (SD) were removed from the graphs. We present the results from both parts of the experiment together. SMALL and BIG are abbreviations referring to the first and second part of the experiment where the menus used were small and big respectively (see Figure 6.1). In both parts, menu ordering (i.e., presentation order of tested menus) did not have any effect on the results.



**Figure 6.2. Visual search, pointing, and total time (+/- standard error) as a function of menu design and level (L1 – level 1, L2 – level 2, and L3 – level 3).**

To determine if there were differences in performance between the menus in the first part of the experiment,  $2 \times 3$  (2 menus  $\times$  3 levels) Two-Way Repeated Measures (RM) ANOVA and post-hoc pair-wise comparisons using Bonferroni adjustment were calculated separately for visual search, pointing, and total (visual search + pointing) time.

Although RM ANOVA is a powerful statistical tool, a set of assumptions have to be met to justify its use. For each dependent measure, and each part separately, we calculated a ratio between the highest and lowest SDs of means from all conditional groups. The maximal ratios varied between 1.6 and 3.1. RM ANOVA is not robust against the violation of assumption of equal variances. In such a situation, it is sensible to transform measured times in order to equalize the variances and obtain times that better approximate normal distribution within conditional groups. We considered log and  $1/x$  transformations. The log one was chosen because it equalized SDs much better.

The assumption of data sphericity was checked with Mauchly's test. When necessary we used the Greenhouse-Geisser correction.

For the data recorded in the second part of the experiment, we used  $4 \times 3$  (4 menus, 3 levels) Two-Way RM ANOVA. The data was log transformed for the same reasons as in the first part of the experiment. Again, all post-hoc pair-wise comparisons used the Bonferroni adjustment, and the sphericity assumption was tested.

### **6.2.1. Visual Search Time**

**SMALL menus.** There was a significant main effect for menu ( $F_{3,51}=78.4$ ,  $p<.001$ ,  $\eta^2=.82$ ), level ( $F_{2,34}=52.8$ ,  $p<.001$ ,  $\eta^2=.76$ ), and menu  $\times$  level interaction ( $F_{6,102}=4$ ,  $p=.001$ ,  $\eta^2=.19$ ). Pair-wise comparisons show that CS (957 ms,  $SD=273$ ) was consistently faster on each level (by 292 ms on average) from all radial alternatives ( $avg=1249$  ms,  $SD=338$ ) which did not differ from each other.

Interestingly, however, while the SPARSE menu is the fastest radial alternative on the first level it becomes slower than the CRL menu on the second level and on the third level it becomes the slowest one. Although this pattern is clearly visible in the data and explains the interaction effect, it is not emphasized by the results of pair-wise comparisons.

Despite the same number of items on all three levels, the visual search times on the second and third levels, although strikingly similar, differ from the first level result (by 130 ms). This pattern is visible for the CRL, FAN, and CS menus. The reason could be the increased effort of having to move the eyes between the levels.

**BIG menus.** There was a significant main effect for menu ( $F_{1,17}=20.9$ ,  $p<.001$ ,  $\eta^2=.55$ ) and level ( $F_{2,34}=274.6$ ,  $p<.001$ ,  $\eta^2=0.94$ ). Times averaged across the levels were 1643 ms ( $SD=572$ ) for CS and 1876 ms ( $SD=618$ ) for CRL. Pair-wise comparisons show that the CS menu was faster than the CRL menu on the second level (by 172 ms) and the third level (by 425 ms) but not the first level.

### **6.2.2. Pointing**

**SMALL menus.** The data was non-spherical for menu x level ( $\chi^2(20)=48.5$ ,  $p<.001$ , G-G  $\epsilon=0.5$ ). There was a significant main effect for menu ( $F_{3,51}=90.3$ ,  $p<.001$ ,  $\eta^2=.84$ ), level ( $F_{2,34}=166.4$ ,  $p<.001$ ,  $\eta^2=.91$ ), and menu x level interaction ( $F_{3,51.3}=6.7$ ,  $p<.001$ ,  $\eta^2=.28$ ). Pair-wise comparisons show that all radial designs (avg=531 ms,  $SD=101$ ) were faster from CS (688 ms,  $SD=129$ ) on all levels and differed from each other only on the third level, where the CRL menu was found to be the fastest. The data shows a clear pattern where the CRL menu gets faster on each level

**BIG menus.** There was a significant main effect for menu ( $F_{1,17}=248$ ,  $p<.001$ ,  $\eta^2=.94$ ) and level ( $F_{2,34}=144$ ,  $p<.001$ ,  $\eta^2=0.89$ ). The comparisons show that the CRL menu (avg=568 ms,  $SD=97$ ) compared to the CS menu (avg=761 ms,  $SD=152$ ) was faster on all three levels (by 193 ms on average).

### **6.2.3. Total time (Visual Search + pointing)**

**SMALL menus.** There was a significant main effect for menu ( $F_{3,51}=10.6$ ,  $p<.001$ ,  $\eta^2=.38$ ), level ( $F_{2,34}=139.5$ ,  $p<.001$ ,  $\eta^2=.89$ ), and menu x level interaction ( $F_{6,102}=7.8$ ,  $p<.001$ ,  $\eta^2=.31$ ). Pair-wise comparisons show that on the first level the SPARSE menu is faster than the FAN menu (by 132 ms). On the second and third levels, times for the SPARSE menu increased the most as a consequence of its increasing visual search time component. On the second level there is no significant difference between the radial menus, but only the CRL menu is not slower than the CS menu (the CS menu being faster than the FAN and SPARSE menus by 173 ms). On the third level again, only the CRL menu is not slower than the CS menu (the CS menu

being faster than the FAN and SPARSE by 250 ms). Additionally, on the third level the CRL menu is faster than the SPARSE menu (by 180 ms).

**BIG menus.** There was a significant main effect of level ( $F_{2,34}=341.8$ ,  $p<.001$ ,  $\eta^2=.95$ ) and menu x level interaction ( $F_{2,34}=5$ ,  $p=.01$ ,  $\eta^2=.23$ ). There was no main effect of menu. The interaction is present because the CRL menu has lower times than the CS menu on the first and second levels but higher on the third level. The pair-wise comparisons, however, show that menus do not differ significantly on the first, second, and third level.

#### **6.2.4. Questionnaires**

**SMALL menus.** We analysed the ranks assigned to menus to reflect ease of finding and selecting items using Friedman ANOVA (nonparametric equivalent of RM ANOVA). One participant completed the questionnaire incorrectly so we excluded it from the analysis. Although participants did not express that finding items was easier with any of the menus ( $\chi^2=5.34$ ,  $df=3$ ,  $p=.15$ ,  $N=17$ ), they subjectively perceived the menus different with respect to easiness of selecting ( $\chi^2=12.44$ ,  $df=3$ ,  $p=.006$ ,  $N=17$ ). Sign Test comparisons between the menus revealed that each radial menu was perceived as faster than the CS menu (CRL-CS  $p=.027$ , SPARSE-CS  $p=.04$ , FAN-CS  $p=.027$ ).

For menu preference orders, we analysed the frequencies with which each menu was chosen first. The Chi-square statistic did not reach significance ( $p=.27$ ). The most popular choice was FAN.

**BIG menus.** For each of the four self-reported variables (see Section 6.1.5) we analysed the frequency with which each condition (CS better, CRL better, no difference) was chosen first by calculating the Chi-square statistic to determine if actual frequencies were significantly different from the case in which all frequencies are equal. Interestingly, the Chi-square statistic did not reach significance for questions related to speed and ease of finding items. However, the participants expressed a strong preference for ease ( $\chi^2(2)=25.3$ ,  $p<.001$ ) and speed ( $\chi^2(2)=13$ ,  $p=.002$ ) of selections, CRL being the most popular choice. There were no significant correlations between any of the variables.

### 6.2.5. Errors

Errors were analysed together for both parts of the experiment. For each participant we aggregated the number of errors (i.e. clicks missing the target) made with each menu. The result of Friedman ANOVA was not significant. A total of 3.6% of all the trials were erroneous.

### 6.2.6. Screen usage

To estimate the screen usage for the radial menus, we calculated a bounding box (and a bounding circle) size that totally enclosed the menu for each level. These two measures are suitable for radial layout if the levels are fully populated. To measure space consumption for smaller menus, we summed the areas of all visible items. For the cascading menus, we summed the sizes of the submenu boxes for each level (see Table 6.1). Screen consumption for the radial menus is lower than for the CS menu by roughly 50% in terms of screen pixels consumed.

**Table 6.1. Screen usage measured in pixels.**

	level 1	level 2	level 3
CS-BIG <sub>PIXELS</sub>	28.800	77.760	146.880
CRL-BIG <sub>PIXELS</sub>	12.168	35.724	74.654
CRL-BIG <sub>BOX</sub>	22.499	77.006	161.202
CRL-BIG <sub>CIRCLE</sub>	19.113	62.458	129.462
CS-SMALL <sub>PIXELS</sub>	24.768	49.536	74.304
CRL-SMALL <sub>PIXELS</sub>	10.647	22.425	36.165

## 6.3. Discussion

### 6.3.1. Visual search and pointing

**Visual search time.** The experimental results show that users searching for commands find them faster in a CS menu than in any radial menu. The advantage is larger for small menus (31%) and smaller for bigger menus (14%). For small menus the CS's 31% advantage is roughly the same on all levels. For big menus the advantage decreases to 10% (138 ms) on the first two levels but reaches 20% (425 ms) on the third level. The CS advantage for visual search is likely due to shorter visual scan paths and parallel visual processing [Hornof and Kieras, 1997]. Using different techniques

for laying out the menu items around the rings (SPARSE, FAN, CRL) does not have any significant effect on search time. However, among the radial menus the CRL menu provides the lowest (and SPARSE the biggest) time difference between first and the third levels.

**Pointing time.** The measurements for the pointing time are reversed: the users perform faster with radial menus than with CS menus with no exceptions – small and big menu sizes and on all three levels. The advantage is similar for two menu sizes (34%). Again, the CRL menu provides the lowest time difference between first and third levels – this time compared to all other menus. This is emphasized by the CRL menu being the fastest menu on the third level. This suggests also that the CRL design shortens the distances between the levels most effectively.

### ***6.3.2. Radial menus (RM) vs CS***

**RM vs CS for novices.** The task used in the experiment simulated novices (i.e., participants always had to search for the items on each level and any learning effects were prevented). Therefore, we can compare novice performance of CS and RM using total selection times. It turns out that the overall selection time for CS and CRL menus do not differ from each other. The CS menu, however, is 10% faster than FAN and SPARSE menus. CRL compensates slower visual search with faster selections. So for novice users, the CS and CRL menus provide similar performance. However, if the selection tasks become more visually demanding than the one used in the current experiment, it is likely that the visual search advantage of the CS menus will dominate the overall selection time.

**RM vs CS for experts.** In contrast to novices, the menu selection performance for experts is a combination of pointing and decision times – search time can be disregarded since experts know the position of menu items. This is based on extensive empirical evidence [Landauer and Nachbar, 1985; Cockburn et. al., 2007] and allows for the following calculations. According to [Hick, 1952; Hyman, 1953; Landauer and Nachbar, 1985; Cockburn et. al., 2007], the decision time is a logarithmic function of the number of items the user has to choose from. Therefore, decision time for the CS and the RM is similar and can be disregarded for a qualitative assessment of CS and radial menus for experts. However, for a quantitative assessment, a decision time offset has to be added to the pointing time. For experts, a decision has to be made before selection on each level [Lane et. al., 1993]. According to [Cockburn et. al.,

2007], the decision times for 6, 7, 12, and 17 items are 447 ms, 465 ms, 527 ms, and 567 ms respectively. When adding the decision time offsets to the pointing times it turns out that the overall selection time is faster by approximately 17% for the RM regardless of the menu level and size (small or big). Our result supports prior evidence where a single level radial menu was found to be 15% faster than linear menu for experts [Callahan et. al., 1988].

**RM vs CS for dedicated users.** Experts have to decide on each level which item to select. However, dedicated users [Card et. al., 1983], who use menus more extensively than experts, need to decide only once at the beginning of the selection which item to select [Lane et. al., 1993]. In this case, they have to choose from all items in the menu. According to [Cockburn et. al., 2007], the decision time for 1428 items (our big menu) is 1078 ms, and for 216 items (small menu) 860 ms. Adding these offsets to the pointing times result in RM being faster by approximately 20% regardless of the menu level and size (small or big).

If the selection tasks become more demanding in terms of pointing than the one used in the current experiment, it is likely that the pointing time advantage of the radial menus will dominate the overall selection time even more. Furthermore, given that the CS menus used in the experiments provide only a lower bound for the required pointing time, the advantage of radial menus are likely to increase if CS menu variants that require precise steering or dwelling are used (see Section 6.1.2).

**Using Point-completion deadline technique (PCD).** The PCD technique is an accepted, legitimate technique used to compare layouts in selection tasks [Hornof, 2001]. However, there is a concern that this technique forces participants into unnatural behaviour since they are forbidden to move the mouse while searching.

*Experts and dedicated users.* Parallel searching and pointing is not of concern for experts and dedicated users since they do not search but point straight to a remembered item location [Landauer and Nachbar, 1985; Kaptelinin, 1993; Cockburn et. al., 2007].

*Novices.* In [Cockburn et. al., 2007] participants were able to move the mouse while searching, but the proposed predictive model assumed that pointing starts after finding an item (as in our study). *"The predictions were accurate within 2% of*

*empirical data*" [Cockburn et. al., 2007] (4 designs tested). A similar result was found in [Byrne, 2001] with the ACT-R/PM model.

Consequently, moving the mouse while searching does not have any practical effect when comparing menu designs.

### **6.3.3. Subjective measurements**

The participants' perception is in agreement with the objective results for novice performance (i.e. no preference for any of the menus) and pointing times (i.e. the radial menus are faster). However, although finding items is faster with the CS menu the participants' perception is that CS and radial menus do not differ.

Participants favoring the radial menus appreciated blank spaces separating the items:

*"Although you can accommodate less items, it seems easier to find/locate items in radial version and the 'white space' gives an uncluttered feeling which leads to a perception of it being easier to use"*

Or

*"It is not convenient going through long lists of items (cascading menu) and the letters mix in the eyes. In the radial menu I was faster and letters are not so mixed since there is blank space between the items".*

Participants in favor of the CS menu stress its predictability and familiarity in contrast to the 'strange' radial layout. This will probably change as users get more familiar with radial menus.

The analysis of the interviews revealed two design considerations we were not aware of. One participant says (big menu size):

*"In radial – the first two levels are very good – better than cascading. The last one is worse than cascading – my eyes needed to do too big moves".*

Some people complain also about distraction caused by the adjacent items from inner levels:

*"the previous levels are 'in the way' geographically of the outer level, causing more confusion when searching for the target as opposed to the cascading layout that produces a*

*visual separation on areas of search (the rectangular areas that contain only the current level)”.*

This suggests that dimming intensity of the inner rings could be a possible improvement for large radial menus.

## **6.4. Conclusions**

This chapter presented a study investigating how radial layout and linear layout influences two main factors governing menu selections – searching for menu items and pointing to menu items.

Visual search is faster with linear layout (CS menu). The advantage is larger for small menus (31%) and smaller for bigger menus (14%). Pointing is faster with radial layout (CRL menu). The advantage is similar for small and big menu size and on all three levels (34%).

The results demonstrate that radial layout is beneficial for the design of displayed hierarchical menus.

The best performing radial menu (CRL menu) did not differ from the CS menu with respect to novice performance. However, if the task becomes more visually demanding than the one used in the current experiment, the CS menu is likely to gain an advantage. This suggests, for example, that using radial layout for menus with changing content may worsen their performance. However, despite slower visual search for the CRL menu, many participants expressed it was easier to search through its ‘not so mixed’ items. Searching the third level ring full of items (big menu) becomes particularly slow for the CRL menu. One potential improvement, in this context, might be to dim the inner rings in order to decrease their visual interference with the outermost ring.

The CRL menu provided the best expert and dedicated user performance for both menu sizes. The CRL menu will be particularly beneficial for professional applications where users invoke hundreds of commands per session (this is also the case for computer games). Some applications may consider using the CRL menu as a quick way of accessing a customized set of commands. If the selection tasks become more demanding in terms of pointing than the one used in the current experiment, it is likely that the advantage of the radial menus will increase. This suggests, for example, that

the CRL menu will gain an additional advantage in menus where multiple selections are possible.

The CRL menu uses less screen space than the CS menu. We recommend using the CRL menu as a context menu to avoid repositioning and slow round-trips to a menu bar. It will be most effective with short command labels (from one to three words).

## ***7. Visual Search and Mouse-Pointing in Radial Menus***

One of the important areas of this research is how novice users search menus. In Section 3.3.5.1 we described that for linear menus it has been established that search is predominantly top-to-bottom. This finding contributes to the understanding of user behaviour with menus, the development of cognitive and menu performance models [Byrne, 2001; Cockburn et. al., 2007], but also to improvements in menu design. Items can be ordered in such a way that the most important items are placed in the top of the menu. Item ordering has notable effect on visual search and pointing times (see Section 3.3.5).

Previous research focused predominantly on linear menus so similar findings regarding menu search are not available for radial menus. The findings regarding visual search in linear menus cannot be readily generalized to radial menus as the layout differences are considerable. Consequently, the question of how to arrange items in radial menus for optimal access remains unanswered.

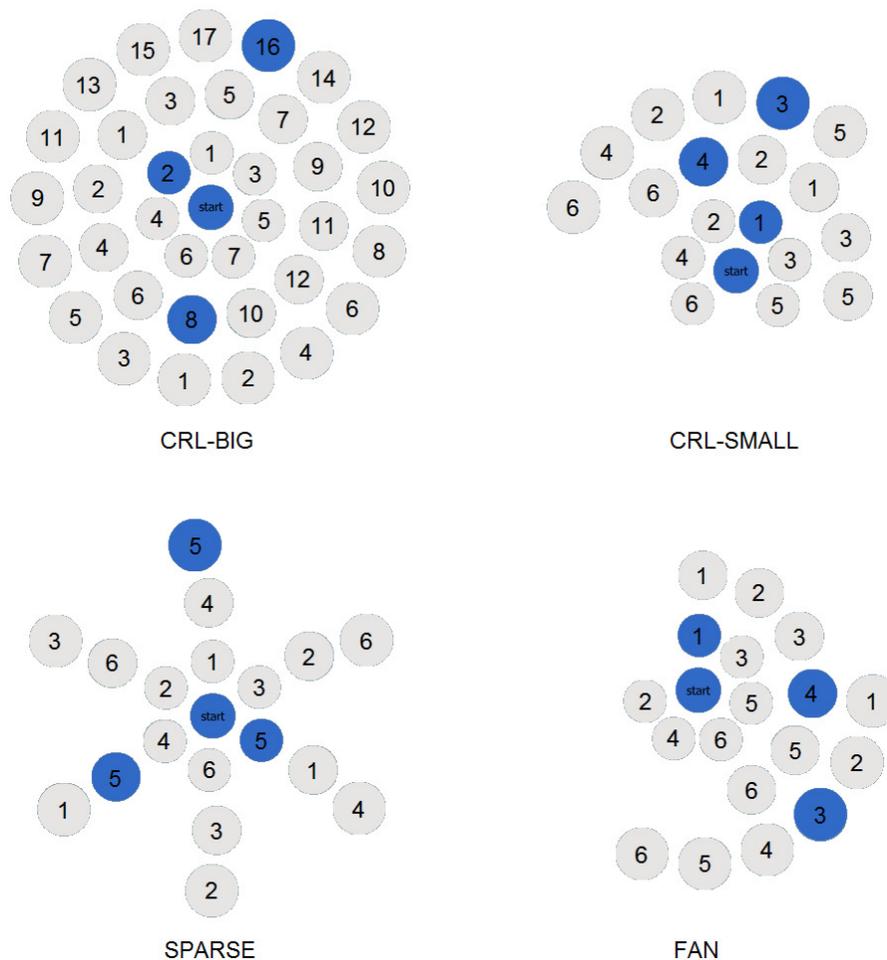
This chapter investigates the order in which users find items in radial menus. We analyse data collected in the experiment presented in Section 6.1. Based on informal analysis we define serial position for items laid out in a circular fashion. For the first level (ring), the serial positions start at 12 o'clock position and alternate between both sides of the ring. For subsequent levels, the serial positions follow distance from a parent item. The defined pattern is tested on empirical data using regression analysis. The defined search pattern yields strong fit and has substantial effect on search performance. We discuss the results in the context of radial menu design.

### ***7.1. Method***

There are a number of plausible strategies for how the items on the rings could be searched – for example, clockwise, counter-clockwise, top-to-bottom or bottom-to-top. The search could start from an item closest to the parent item or from an item on any

end of an arc (see Figure 6.1 and Figure 7.1, FAN and CRL-SMALL designs). Search might also lack any systematic patterns and be best described as random. Note that different search strategies result in different items being found first. Therefore, it is not clear how to arrange items for optimal menu access and it is necessary to analyse empirical data.

We use the data collected in the experiment presented in Section 6.1. The experiment employed the four radial menus shown in Figure 7.1.



**Figure 7.1. Radial menus used in the experiment presented in Chapter 6. The labels have been removed and substituted with serial positions. The selected items are coloured blue.**

Visual search time was separated from pointing time using the Point-Completion Deadline (PCD) technique described in [Hornof, 2001]. In short, the technique discourages participants from moving the mouse until they found the target.

Consequently, the start of the mouse movement consistently marks the end of visual search. The PCD is an accepted, legitimate technique used to reliably separate and measure visual search times in selection tasks [Hornof, 2001; Burke et. al., 2005; Samp and Decker, 2010]. This technique, however, does not employ eye-tracking. Therefore, what we focus on in this chapter is not patterns of the actual eye movements but rather patterns identifying **order of finding** the items. Such patterns (i.e., the order in which items are found) would inform design of radial menus as to how order the items for optimal search, would shed some light at how users search radial menus, and would help formulate hypotheses for more detailed eye-tracking studies.

Chapter 6 focused on average times for each menu design and each menu level. Therefore, it is not known how position on the ring affected measured times or which items were found faster. This chapter focuses on menu designs, levels, and most importantly on individual positions on a ring.

To find regularities in how participants search the rings, we aggregated visual search time across participants using an arbitrarily chosen but consistent scheme of numbering the items around the rings. This was done for each menu design and each menu level separately. Ordering the items according to the visual search times led to the observation that for the second and third levels, search time seemed to be a function of the distance from a parent item. For the FAN design this results in search following the ring clockwise starting from the item closest to the parent. For the CRL-BIG, CRL-SMALL, and SPARSE designs the result is search alternating between both sides of the ring (i.e., half rings on both sides of the parent item), again starting from the item closest to the parent (consult with Figure 7.1). For the first level, where all the items are equally distant from the center, we observed that search starts from the top item (i.e., the item at 12 o'clock) and also alternates between both sides (in this case left and right) of the ring.

## **7.2. Results**

### **7.2.1. Analysing general search pattern**

Consistent with our observations, we define serial position of the items on the first ring in an alternating manner starting from the top item (i.e., the item at 12 o'clock). For all subsequent rings, serial position of the items is defined according to their distance from a parent item. Figure 7.1 superimposes serial positions on the items of the four

radial menus. From now on, whenever we say *the first* or *the last items*, we refer to the order defined by the serial position.

We performed regression analysis to measure the actual relationship between the serial position and the visual search time/pointing time. For each menu design and serial position, times were aggregated across participants and levels. This allowed us to obtain a number of observations necessary for establishing reliable mean time per serial position. Moreover, it allowed us to focus on the general pattern of which items are found faster regardless of menu level. If such a general pattern exists, it will be more readily applicable in practice compare to, for example, different patterns for different levels. We also performed a separate regression on times aggregated only across small radial menus. The regression results are presented in Figure 7.2.

Note that the aggregation across levels could flatten the possible effect of the level, especially for the big radial menu, where each level contains a different number of items. To this end, we also analyse the effect of menu size later in the chapter.

### **7.2.1.1. Visual Search as a function of serial position**

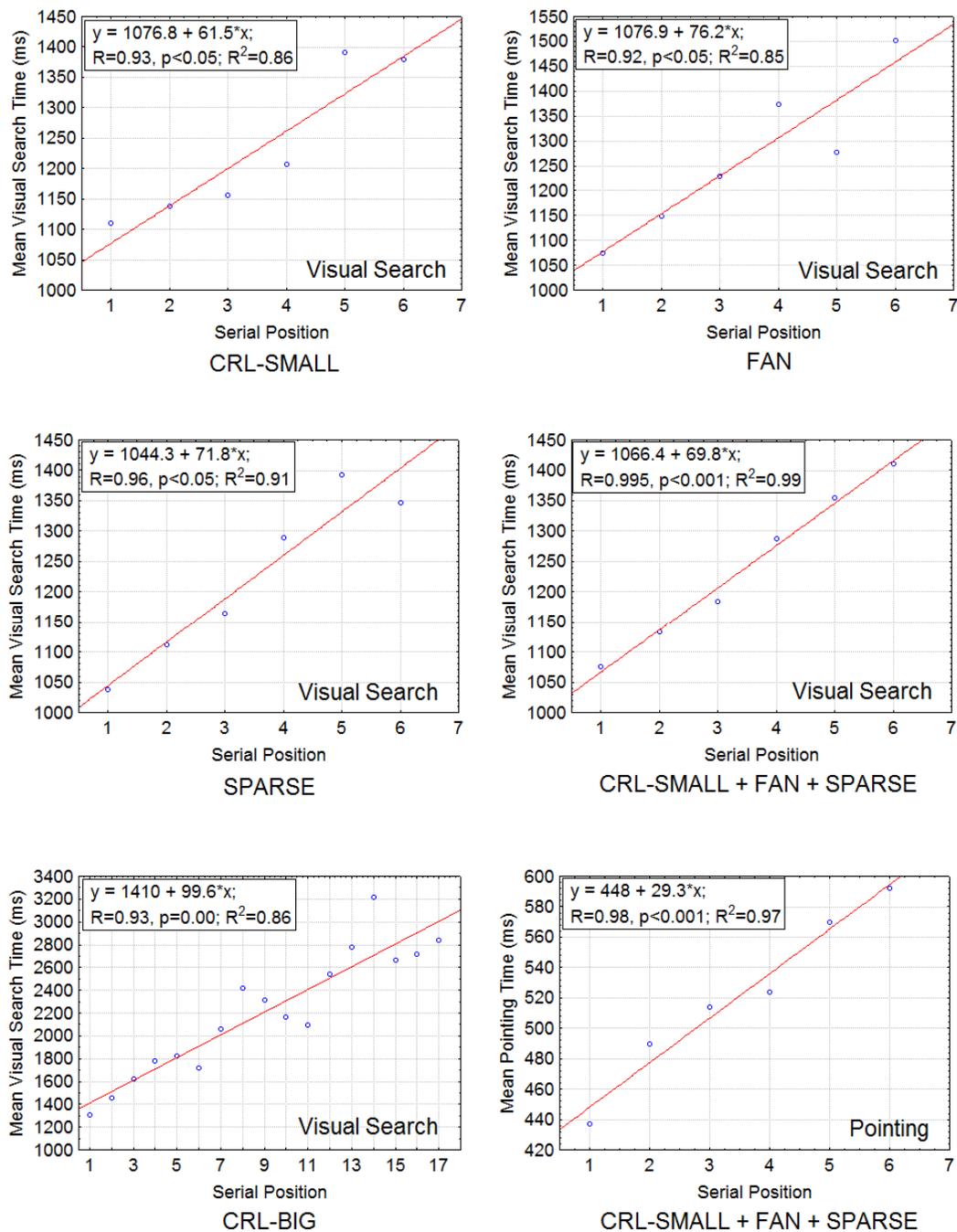
The average number of observations per serial position was 133 (SD 30) for small radial menus (CRL-SMALL, SPARSE, FAN) and 97 (SD 60) for big radial menu (CRL-BIG). The results are shown in Figure 7.2. The fit is strong across menu designs and two menu sizes ( $R^2 > .85$  and  $R^2 = .99$  for times aggregated across three small radial menus). The results indicate that the defined search pattern holds for all menu designs used in the experiment. Interestingly, search that alternates between both sides of the ring for the CRL-SMALL, CRL-BIG, and SPARSE designs (being a function of the distance from a parent item) leads to a non-optimal scanning path in terms of its total length. Search following circumference along one direction (e.g., clockwise starting in any position) would have resulted in a shorter path. We hypothesize that, when unable to find a target item in a given location, participants felt encouraged to move to an entirely different location.

Regression lines are fairly similar across small radial menus (intercepts ranging from 1044 ms to 1077 ms and slopes ranging from 61 ms to 76 ms). Each subsequent serial position is associated with an average 70 ms increase in visual search time. The regression line for the big radial menu has a notably larger intercept (1410 ms) and slope (100 ms) when compared with the results for the small menus. For the big menu, each subsequent serial position is associated with an average increase of visual search

time of 100 ms. This indicates that the menu size (length) affects the search performance. We explore this in the next section.

Figure 7.2 shows that the difference between finding the first and the last item is substantial. For small radial menus, this difference is on average 360 ms (approximately 30% of the average visual search time for small menus). For the big radial menu the difference is 1900 ms (approximately 85% of the average visual search time for the big menu). Consequently, the arrangement of items on the rings can lead to notable performance differences.

Note also that the defined serial position will also have a linear effect on the pointing time since each subsequent item (according to serial position) is more distant from a parent item and thus slower to select according to Fitts' law [Fitts, 1954]. We run regression analysis of pointing times against serial position (see the last graph in Figure 7.2). As expected, the fit is strong ( $R^2=.97$  for times aggregated across the three small radial menus, and  $R^2=.81$  for the big radial menu). Restricted by the available space, we did not include a graph for the big radial menu. Its lower correlation may be partially attributed to a smaller number of observations available for that menu. For small radial menus the difference between selecting the first and the last item is 160 ms (approximately 30% of the average pointing time for small menus). For the big radial menu, the difference is 300 ms (approximately 48% of the average pointing time for the big menu). Consequently, arranging items according to the proposed pattern (i.e., most important items in the first serial positions) will improve not only search time but also pointing to these items which is important for both novices and experts [Cockburn et. al., 2007].



**Figure 7.2. Visual Search Time and Pointing Time as a function of serial position. The best-fit regression lines and corresponding statistics are provided for each condition. Notable dispersion of the visual search times after the seventh item for the CRL-BIG menu can be partially attributed to the smaller number of observations available for those item positions.**

### 7.2.1.2. Visual Search as a function of menu size

The big radial menu (CRL-BIG) has different number of items on each level (7, 12, 17). This allowed us to measure the effect of menu size on visual search time much like [Nilsen, 1991; Sears and Shneiderman, 1994] did for linear menus. We aggregated times for the first seven items on each level and performed regression between these times and the three menu sizes. Regression analysis produced a strong model with  $\text{SearchTime} = 140 * \text{MenuSize} + 209$ ,  $R^2 = .99$ . The visual search time increased by approximately 700 ms between the analysed menu sizes<sup>4</sup>. Figure 7.3 shows the results.

Menu size has a similar linear effect on visual search time for radial menus as it has for linear menus. However, the size of the effect is notably larger for radial menus. Our results indicate that each additional item in a radial menu increases the average visual search time by 140 ms. Cockburn et al. [Cockburn et al., 2007] report an 80 ms increase for each additional item in a linear menu. The difference is considerable. It would be interesting to explore which parameters of radial menus – for example, the circular trajectory, item density, item spacing, item shapes – have the strongest effect on the slowdown.

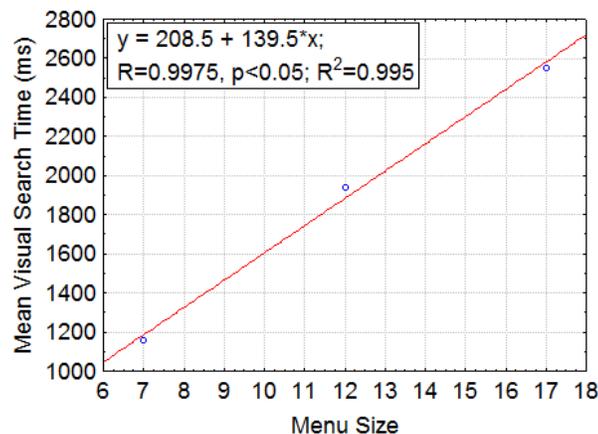


Figure 7.3. Visual Search Time as a function of menu size.

### 7.2.1.3. Mouse pointing as a function of serial position

The fit for the pointing time is strong ( $R^2 = .97$  for times aggregated across three small radial menus, and  $R^2 = .81$  for the big radial menu). The lower correlation for the big

<sup>4</sup> We also performed the same analysis for four menu sizes (including size=6 from small menus). The results were almost identical,  $R^2 = .98$ .

radial menu may be partially attributed to the smaller number of observations available for that menu.

The strong fit is not surprising since larger serial positions are more distant and thus slower to select according to Fitts' law [Fitts, 1954]. For the small radial menu the difference between selecting the first and the last item is 160 ms (approximately 30% of the average pointing time for the small menu). For the big radial menu, the difference is 300 ms (approximately 48% of the average pointing time for the big menu).

### **7.2.2. Analysing individual search patterns**

Following our investigation, we analysed individual search patterns. For each participant, the menu design, and serial position, times were aggregated across the levels and plotted as a function of serial position (72 plots). We made two observations analysing the individual plots. First, sometimes participants searched not one, but two or three consecutive items before moving to the opposite side of the ring. Second, participants employed not one but various search strategies, of which the following were dominant: 1) the search pattern described in the chapter; 2) searching in an alternating manner but starting with the last items; 3) starting with the first items and searching half of the ring to the left/right before proceeding to the other half. Some searches could best be described as random. Note that for some of these strategies the items with the first serial positions were actually found last<sup>5</sup>. For this reason, the pattern reported in this chapter should not be treated as a definitive order in which participants find items on the ring but rather as general regularity (much like top-to-bottom regularity in linear menus [Byrne et. al., 1997; Aaltonen et. al., 1998]). In other words, the pattern indicates which items are found faster and slower in a statistical sense – i.e., on average.

For the same reason, and to better understand strength of the proposed pattern, we decided to extend our analysis and check how often the first items are actually found faster. To this end, for each participant and menu design, we aggregated visual search time for each serial position. Further, for each participant and design, we counted how many items from the first half of all the items (according to our serial position) a participant found faster than the half of all the items (i.e., faster than the majority of

---

<sup>5</sup> We tried alternative ordering schemas but they resulted in a poor fit. For practical reasons and space wise we focused on the prevailing search pattern.

the items). For each participant we average this count across the menus to obtain one count per participant – we call this count TOP-IN-TOP (the *top* half of the items in the *top* half of the measured times). Similarly, we counted how many items from the first half of all the items a participant found slower than the half of all the items. For each participant, the TOP-IN-BOTTOM (the *top* half of the items in the *bottom* half of the measure times) is the average of this count across the menus. TOP-IN-TOP and TOP-IN-BOTTOM, divided by their sum, represent the probabilities of finding an item from the first half of the items faster/slower than a half of all the items (in other words faster/slower than the majority of the items). The distributions of TOP-IN-TOP and TOP-IN-BOTTOM measures were normal (the Shapiro-Wilk's *W* test was used,  $p > 0.05$ ). We performed the dependent measures t-test to determine if TOP-IN-TOP and TOP-IN-BOTTOM differed significantly.

For small radial menus, the mean number of TOP-IN-TOP items was 2 (SD 0.58) (associated probability 67%) and the mean number of TOP-IN-BOTTOM items was 1 (SD 0.58) (probability 33%). For the big radial menu, the mean number of TOP-IN-TOP items was 6.5 (SD 1) (probability 72%) and the mean number of TOP-IN-BOTTOM items was 2.5 (SD 1) (probability 28%).

The results of the t-test show that, indeed, the first half of items determined by our serial position are found faster than the majority of items ( $p < .05$  for small radial menus, and  $p < .01$  for the big radial menu). This reinforces the results obtained in the regression analysis. However, the reported means and probabilities demonstrate that the search pattern determined by the serial position is not as strong as could be asserted solely on the basis of correlation score. In other words, other search patterns are also important.

In sum, placing the most important items according to the proposed pattern gives an approximately 70% chance for these items to be found faster than the majority of the items. This is 20% better than a random order. Given substantial differences between visual search time for the items found the fastest and the slowest (30-85%, see Section 7.2.1.1), following the proposed pattern will lead to better overall performance of radial menus. This is important as searching radial menus is known to be slower than searching linear menus [Samp and Decker, 2010].

The existence of other patterns hints that designers might consider solutions that suggest the order of reading, such as using an arrow or using animation which reveals the items in a desired order.

### **7.3. Conclusions**

In this chapter we investigated the order in which users find items in radial menus. We found that for the second and third levels, search time is a function of the distance from a parent item. For the first ring, where all the items are equally distant from the center, search alternates between both sides of the ring, starting from the top item. The difference between finding the first and the last item is substantial: 30% and 85% of the average search time in small and big radial menus respectively. Similarly, the difference between pointing to the first and to the last item is substantial: 30% and 48% of the average pointing time for small and big menus respectively. Given that the search and pointing are the dominant components of menu selection times [Cockburn et. al., 2007], the order of the items on a ring will affect menu performance.

Our results suggest that the items a designer wishes to be found faster (e.g., the most important items) should be placed in the top of the first ring and in the positions closest to the parent item for all subsequent levels. Users will also benefit from decreased pointing times to these items.

## **8. Conclusions and Future Work**

This thesis began with an observation that graphical menus provide slow performance and yet users often try to select menu commands as fast as possible. We saw that shortcut techniques that enable users to by-pass selections from a graphical menu are not the ultimate answer to the performance problem. Shortcut techniques pose high demands on human memory. They require much training and frequent use. They do not scale well. And they often cannot be provided on devices with limited capabilities, such as mobile devices with small screens, small keyboards, and touch screens. Because of these problems, users often times ‘stick’ to a graphical menu and are condemned to its poor performance.

We argued then that it is possible to build efficient graphical menus if performance is an explicit design goal. In Chapter 3 we presented a design of such a menu: the CRL menu. It was designed with both novices and experts in mind. The design sought efficiency predominantly through the use of a concentric radial layout which decreases navigation distances and increases item sizes. Further chapters evaluated the menu and its characteristics in controlled experiments. The results lead to a number of conclusions and directions for the future work which we divide into two categories. The first category (Section 8.1) relates to the CRL menu design, its performance, and the fulfilment of the central hypothesis of this thesis. The second category (Section 8.2) relates to more general HCI knowledge, beyond graphical menus.

### **8.1. CRL menu**

The following list summarizes results from the studies:

- Chapter 4 showed that the CRL menu provides 34% better *navigation performance* than a cascading menu for three-level deep command hierarchies. Participants also perceived the CRL menu as being faster.
- Chapter 4, through modelling, also showed that, compared to the cascading menu, the CRL menu: 1) considerably decreased the difficulty (ID) of structurally equivalent tasks; and 2) biased participants towards speed,

requiring less accuracy. Consequently, apart from navigating faster, CRL menu users might also be able to navigate with less attention and mental fatigue.

- Chapter 5 showed that the CRL menu leads to fewer navigation errors than the cascading menu, both quantitatively and qualitatively. This is mainly because the CRL menu does not restrict navigation paths, does not require a steering task, and enlarges the size of menu items.
- Chapter 6 showed that the CRL menu is 34% faster than the cascading menu in terms of pointing time but that at the same time it is 14%-31% slower in terms of visual search time. Interestingly, although participants perceived the decrease in pointing time they did not perceive the increase in visual search time.
- Chapter 6 showed that the CRL menu and the cascading menu did not differ in terms of *novice performance*. The CRL menu counterbalances slower search with faster selections.
- Chapter 6 showed that the CRL menu provides the best *expert performance* in all tested conditions, 17%-20% better than that of the cascading menu.
- Chapter 6 demonstrated that the CRL menu consumes less screen space than the cascading menu.
- Chapter 7 found a pattern of item placement for radial layout that can improve visual search times compared to a random order. The items a designer wishes to be found faster (e.g., the most important items) should be placed in the top of the first ring and in the positions closest to the parent item for all subsequent levels.

In Section 1.2 we established the goal of this thesis:

*“The central hypothesis of this work is that graphical menus can be designed with performance in mind, and that such graphical menus can perform better than the most widely used cascading menus. Improved performance has many dimensions including shortened selection times, decreased navigation difficulty, lowered error rates, and inducing users to fast movements that require little precision”.*

The design presented in Chapter 3 and the above results support our central hypothesis, quantitatively and qualitatively. The CRL menu results in approximately 34% faster navigation, up to 20% better performance for expert users, decreases the difficulty of selection tasks, lowers error rates, and biases users toward speed and less

accuracy. These benefits hold across different menu sizes. The CRL menu will be particularly beneficial for professional applications where most users are experts (or quickly become experts) and invoke hundreds of commands per session (this is also the case for computer games). If the selection tasks become more demanding, in terms of pointing, than the one used in the experiment, it is likely that the advantage of the CRL menu will increase. This suggests that the CRL menu will gain an additional advantage in menus where multiple selections are possible.

We recommend using the CRL menu as a context menu (i.e., a menu posted on request and appearing close to the current cursor position) to avoid repositioning and slow round-trips to a menu bar. Some applications may consider using the CRL menu as a quick way of accessing a customized set of commands, those accessed most frequently. The CRL menu is an interesting option for devices with small screens, such as mobiles and tablets, because it uses less screen space than the cascading menu. It might be also a preferred option for touch interactions because the circular item shapes better match finger tips compared to the thin rectangles used by the cascading menu. This property might be particularly beneficial since many touch devices are used on the move where interactions might be more inaccurate. Future work could test the CRL menu, and its tolerance to inadequate interactions, on handheld touch devices in scenarios where participants are on the move (e.g., walking up the stairs).

The CRL menu did not differ from the cascading menu with respect to novice performance. However, if the task becomes more visually demanding than the one used in the experiment, the visual search time differences will play more important role, and the cascading menu is likely to gain an advantage. This suggests, that the CRL menu might become slower than the cascading menu for novices if the menu content is dynamic and changes frequently (e.g., as a result of adaptation). Searching the third level ring full of items becomes particularly slow for the CRL menu. Participants complained that the inner rings ‘get in their way’ while searching the outer rings. One potential remedy might be to dim the inner rings in order to decrease their visual interference with the outermost ring. Interestingly, despite slower visual search for the CRL menu, many participants expressed that it was easier to search through its ‘not so mixed’ items.

Future work on radial menus could focus on measuring the effectiveness of some specific features such as *menu area*, *item prediction* (see Section 3.4.6.2), those

dealing with limited space on a ring (see Section 3.4.1), and *item grouping* (see Section 3.4.5).

## **8.2. Radial layout in Graphical User Interfaces**

Early in this thesis we observed that Graphical User Interfaces were historically created to introduce computers to a broad audience, especially users with little or no computing expertise. And as such, they were not designed with performance in mind. Our design and evaluation demonstrates, however, that if performance is an explicit design objective, Graphical User Interfaces can become faster and less error-prone – that is, more suitable for expert use. There exists a rich body of research that can inform design decisions when seeking to improve selection performance. Radial layout is just one example of achieving this goal and graphical menus are just one example of a Graphical User Interface. We believe that the design goals and characteristics presented in Section 3.2 and Section 3.3 can be successfully applied to the design of other types of Graphical User Interfaces. Future work could focus on the design or re-design of other Graphical User Interfaces which are used frequently. It would be also useful to extend the list of design characteristics from Section 3.3 with other characteristics that directly benefit or impact performance.

The results presented in this thesis demonstrate that radial layout is beneficial for organizing frequently accessed graphical interface elements. At the same time, the results demonstrate that finding items on the rings is slower than on lists. This finding is perhaps not surprising given long prevalence of list organization in human history. Think of lists used to organize names in phone books, terms in book indexes, and entries in restaurant menus. Our familiarity with list organization is also reinforced by the way we read and write texts: top-to-bottom. Consequently, if people are slower finding items on the rings we should be careful with applying radial organization to browsing and exploratory interfaces. Exploration and browsing employ much visual search and lists might better guide users through this process. One thing we learnt about linear vs radial organization is that while a list has a clear beginning and end, the same cannot be said about a ring. This might affect strategies of exploratory search, particularly the ability to scan all alternatives and keep track of visited and unvisited locations. Future work could explore different ways, and their effectiveness, for suggesting direction/strategy of visual search on the rings. Both temporal cues (i.e., animation) and graphical cues (e.g., round arrows) can be considered.

However, we believe that radial layouts can benefit browse and exploratory interfaces in scenarios where the spatial characteristics of a radial layout can be leveraged. A radial layout, compared to a linear one, expands in all directions. This can help structure an information space and guide user actions, especially if the task or data is inherently spatial. For example, items representing building floors can be organized in a radial interface in such a way that the interface levels expand upwards while a user explores higher floors and expand downwards for lower floors. Another use of radial layouts can take advantage of the fact that items surrounding the center communicate their strong relationship to the central item. A user exploring a car architecture diagram can click on different components (e.g., a wheel, a mirror, and a bumper) and see the components' parts organized in a radial fashion, surrounding the selected component. Another use of spatial characteristics of radial layout is to organize items in such a way that the items close on a ring are highly related while the ones on the both ends of a diameter represent the opposite or non-related items. Yet another example is to organize data by hours which are laid out along the ring like hours on a wall clock. In the above examples, although scanning rings is slower than scanning lists, browse and exploration strategies could potentially leverage the spatial characteristics of radial layout. Future work could investigate these issues. It would be also interesting to test if and how direction serves as an additional discrimination feature while users learn the structure of the information space.

This thesis employed the Point Completion Deadline technique to measure visual search times in radial menus. Consequently, we were able to identify search patterns based on how fast individual items were found. Such patterns are useful for improving designs but they tell us very little about the actual user behaviour. Future work could use eye tracking equipment to reveal users' actual eye movements and their actual search strategies. Results presented in this thesis can help to formulate initial hypotheses for such studies.

This thesis also demonstrated that there is more involved in navigation errors than the number of clicks that miss a target item. Our NTV measure showed that users had more problems navigating the cascading menu. This finding was supported by subjective feedback. The proposed measure, however, was not systematically validated. Future work should test, in a controlled setup, if and how well the NTV measure captures different navigation problems of varying severity.

An interesting finding was the lack of subjective support for objectively faster visual search in the case of the linear layout. Users often complained about the close proximity of items in the linear layout and appreciated spacing between the items in the CRL menu. This suggests that there is more involved in visual search than performance. Future work could explore other properties – particularly item size and spacing – of visual search that influence user subjective perception and search comfort. We found also that for radial layout visual search is particularly slow on the third ring. Participants complained that the inner rings ‘get into their way’ while searching the outer rings. Future work should investigate the effects of visual interference on visual search performance and user behaviour.

Our modelling work showed that hierarchical navigation, apart from pointing, consists of sub-selection additive factors whose effects are substantial. Future work on modelling could test these effects more thoroughly and incorporate them in menu performance models.

Finally, we would like to express our hope that the work presented in this thesis will encourage practitioners and researchers to work on Graphical User Interfaces that are designed with both novices and experts in mind.

### **8.3. Outcomes**

The work presented in this thesis led to a number of outcomes; often beyond those required by the thesis. We list them here.

- 1) In 2007 the CRL menu design was highly commended (three highly commended among 63 submissions) at the Summer School on Multimedia Semantics in Glasgow, Scotland.  
<http://www.dcs.gla.ac.uk/ssms07/>
- 2) The CRL menu has been presented at the Semantic Technology Conference in 2008, San Jose, USA as a poster and during a full talk entitled “Choosing the Best UI for Your Semantics”. The CRL menu has been also presented at the Semantic Web User Interface workshop at CHI 2008, Florence, Italy, at the IEEE International Conference on Digital Information Management, London, UK, at the Irish Human-Computer Interaction Conference, 2008, Cork, Ireland, and at BigIdeas Technology Showcase in Dublin, Ireland.

- 3) In 2010 a paper entitled “Supporting menu design with radial layouts” was accepted to the Advanced Visual Interfaces (AVI), Rome, Italy. Chapter 6 was partly based on this paper. The paper contributed to HCI research on spatial layouts and their influence on two components of completion time: visual search time and pointing time. The paper also contributed a comprehensive comparison of four layouts including their novice and expert menu performance.
- 4) In 2011 a paper entitled “Navigation Time Variability: Measuring Menu Navigation Errors” was accepted to the Conference on Human-Computer Interaction, INTERACT, Lisbon, Portugal. Chapter 5 was partly based on this paper. The paper contributed a new measure of navigation errors, which addresses the shortcomings of current error rate measures, to HCI research. The paper also contributed a comparison of the CRL and the CS menus using the proposed measure.
- 5) In 2011 a paper entitled “Visual Search in Radial Menus” was accepted to the Conference on Human-Computer Interaction, INTERACT, Lisbon, Portugal. Chapter 7 was partly based on this paper. The paper contributed to the HCI research on visual search in menus and user interfaces. The paper contributed a pattern which optimizes visual search and pointing times in radial layouts.
- 6) Three versions of the CRL menu have been developed as web components and made free for non-commercial use.  
<http://krystiansamp.com/atominterface>.
- 7) The second and third version of the CRL menu have been commercialised by the University Technology Transfer Office. Four licensing agreements have been signed. The CRL menu is already used in production by MyKidsTime.ie (<http://www.mykidstime.ie>).
- 8) In 2011 an article on commercial success of the CRL menu appeared in the Irish Times, printed and digital format  
<http://www.irishtimes.com/newspaper/finance/2009/0911/1224254276728.html>
- 9) Some features of the CRL menu design presented in Chapter 3 resulted in an international patent application, filed in 2009 (Pub. No.: WO/2010/007485, International Application No.: PCT/IB2009/006192). The features address the limitation of space on the rings and on the screen in radial layouts. In 2011 a US application was filed.  
<http://www.wipo.int/pctdb/en/wo.jsp?WO=2010007485>



## References

- Aaltonen, A., Hyrskykari, and A., Rähkä, K.-J., 1998. 101 spots, or how do users read menus?. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 132-139.
- Accot, J. and Zhai, S., 1997. Beyond Fitts' Law: Models for trajectory-based HCI tasks. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, New York, ACM, 295-302.
- Adam, J.J., 1992. The effects of objectives and constraints on motor control strategy in reciprocal aiming movements. In *Journal of Motor Behavior*, 24, 173-185.
- Adam, J.J., and Paas, F.G.W.C., 1996. Dwell time in reciprocal aiming tasks. In *Human Movement Science*, 15(1), 1-24.
- Ahlström, D., 2005. Modeling and improving selection in cascading pull-down menus using Fitts' law, the steering law and force fields. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 61-70.
- Ahlström, D., Alexandrowicz, R., and Hitz, M., 2006. Improving menu interaction: a comparison of standard, force enhanced and jumping menus. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 1067-1076.
- Ahlström, D., Cockburn, A., Gutwin, C., and Irani, P., 2010. Why it's quick to be square: modeling new and existing hierarchical menu designs. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 1371-1380.
- Anderson, J.R., Matessa, M., and Lebiere, C., 1997. ACT-R: a theory of higher level cognition and its relation to visual attention. In *Human-Computer Interaction*, 12, 439-462.
- Anderson, J.R., Matessa, M., and Lebiere, C., 1998. The visual interface. In *The Atomic Components of Thought*, Mahwah, NJ: Lawrence Erlbaum, 143-168.
- Asano, T., Sharlin, E., Kitamura, Y., Takashima, K., and Kishino, F., 2005. Predictive interaction using the delphian desktop. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, 133-141.
- Augustyn, J.S., and Rosenbaum, D.A., 2005. Metacognitive control of action: Preparation for aiming reflects knowledge of Fitts's law. In *Psychonomic Bulletin & Review*, 12(5), 911-916.

- Bailly, G., 2007. Wave Menus: Improving the novice mode of Marking Menus. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT)*, 475-488.
- Bau, O., and Mackay, W.E., 2008. OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the annual ACM symposium User Interface Software and Technology*, 37-46.
- Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A., 2003. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch and pen-operated systems. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT)*, 57-64.
- Beaudouin, M.L., 2004. Designing interaction, not interfaces. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)*, 15-22.
- Bederson, B.B., 2000. Fisheye menus. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, 217-225.
- Bekkering, H., Adam, J.J., Kingma, H., Huson, A., and Whiting, H.T.A., 1994. Reaction time latencies of eye and hand movements in single- and dual-task conditions. In *Experimental Brain Research*, 97(3), 471-476.
- Bernard, M.A., 1994. On the Automated Drawing of Graphs. In *Proceedings of Caribbean Conferences on Combinatorics and Computing*, 43-55.
- Bernard, M., and Mohammed, S., 2003. Labeled Radial Drawing of Data Structures. In *IEEE Symposium on Information Visualization (INFOVIS)*, 479-484.
- Bhavnani, S.K., Flemming, U., Forsythe, D.E., Garrett, J.H., Shaw, D.S., and Tsai, A., 1996. CAD usage in an architectural office: from observation to active assistance. In *Automation in Construction*, 5(3), 243-255.
- Black, J.B., and Moran, T.P., 1982. Learning and remembering command names. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 8-11.
- Blanch, R., Guiard, Y., and Beaudouin-Lafon, M., 2004. Semantic pointing: improving target acquisition with control display ratio adaptation. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 519-526.

- Bock, O., and Eversheim, U., 2000. The mechanisms of movement preparation: A precuing study. In *Behavioural Brain Research*, 108, 85-90.
- Bohan, M., Longstaff, M.G., Van Gemmert, A.W.A., Rand, M.K., and Stelmach, G.E., 2003. Differential effects of target height and width on 2D pointing movement duration and kinematics. In *Motor Control*, 7, 278-289.
- Brooks, F.B., 1995. *The Mythical Man-Month*, Addison-Wesley, Reading, MA.
- Buck, L., 1986. Target location effects in tapping tasks. In *Acta Psychologica*, 62, 1-13.
- Burke, M., Hornof, A.J., Nilsen, E., and Gorman, N., 2005. High-cost banner blindness: Ads increase perceived workload, hinder visual search, and are forgotten. In *ACM Transactions on Computer-Human Interaction (ToCHI)*, 12(4), 423-445.
- Byrne, M.D., 1993. Using icons to find documents: simplicity is critical. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT)* and in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 446-453.
- Byrne, M.D., Anderson, J., Douglass, S., and Matessa, M., 1997. Eye Tracking the Visual Search of Click-Down Menus. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 402-409.
- Byrne, M.D., 2001. ACT-R/PM and menu selection: applying a cognitive architecture to HCI. In *Interaction Journal of Human-Computer Studies*, 55, 41-84.
- Calfee, R., 1975. Associative learning. In *Journal of Experimental Psychology: Human Experimental Psychology*, 379-399.
- Callahan, J., Hopkins, D., Weiser, M., and Shneiderman, B., 1988. An empirical comparison of pie vs. linear menus. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 95-100.
- Card, S.K., English, W.K., and Burr, B.J., 1978. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. In *Ergonomics*, 21(8), 601-613.
- Card, S.K., Moran, T.P., and Newell, A., 1980. The keystroke-level model for user performance time with interactive systems. In *Communications of the ACM*, 23(7), 396-410.

- Card, S.K., 1982. User perceptual mechanisms in the search of computer command menus. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 190-196.
- Card, S.K., Moran, T.P., and Newell, A., 1983. *The Psychology of Human-Computer Interaction*. LEA.
- Carroll, J., 1984. Minimalist design for active users. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT)*, 39-44.
- Carroll, J.M., and Mazur, S.A., 1985a. LisaLearning. IBM Research Report.
- Carroll, J.M., Mack, R.L., Lewis, C.H., Grischkowsky, N.L., and Robertson, S.R., 1985b. Exploring exploring a word processor. In *Human-Computer Interaction*, 1(3), 283-307.
- Carroll, J., and Rosson, M., 1987. Paradox of the active user. In *Interfacing thought: Cognitive aspects of human-computer interaction*, 80-111.
- Carroll, J., 1990. The nurnberg funnel: Designing minimalist instruction for practical computer skill. In *MIT Press*.
- Charney, D., Reder, L., and Kusbit, G., 1990. Goal setting and procedure selection in acquiring computer skills: a comparison of problem solving and learner exploration. In *Cognition and Instruction*, 7(4), 323-342.
- Chua, R., Weeks, D.J., and Goodman, D., 2003. Perceptual-motor interaction: some implications for human-computer interaction. In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, 2<sup>nd</sup> Edition, 27-42.
- Cockburn, A., and Gin, A., 2006. Faster cascading menu selections with enlarged activation areas. In *Proceedings of the Graphics Interface Conference (GI)*, 65-71.
- Cockburn, A., Gutwin, C., and Greenberg, S., 2007. A predictive model of menu performance. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 627-636.
- Conklin, J., 1987. Hypertext an introduction and survey. In *IEEE Computer*, 20(9), 17-41.

- Coren, S., and Girgus, J.S., 1980. Principles of perceptual organization and spatial distortion: The gestalt illusions. In *Journal of Experimental Psychology: Human Perception and Performance*, 6(3), 404-412.
- Cuthbert, P., 1992. Concepts not keystrokes - Can we improve our teaching of word processing?. In *Computer Education*, 72, 2-5.
- Davis, S., and Bostrom, R., 1992. An experimental investigation of the roles of the computer interface and individual characteristics in the learning of computer systems. In *International Journal of Man-Machine Studies*, 4, 143-172.
- De Alwis, B., and Murphy, G.C., 2005. Remaining oriented during software development tasks: An exploratory field study. *Technical Report in Department of Computer Science in the University of British Columbia*.
- Dean, M., Wu, S.W., and Maloney, L.T., 2007. Trading off speed and accuracy in rapid, goal directed movements. In *Journal of Vision*, 7(5):10, 1-12.
- Dennerlein, J.T., Martin, D.B., and Hasser, C., 2000. Force-feedback improves performance for steering and combined steering-targeting tasks. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 423-429.
- Dieberger, A., 1997. Supporting social navigation on the World Wide Web. In *International Journal of Human-Computer Studies*, 46(6), 805-825.
- Dix, A., 1995. Accelerators and toolbars: learning from the menu. *People and Computers IX (Adjunct): British Computer Society Conference on Human Computer Interaction (HCI)*, 138-143.
- Dodge, R., 1907. An experimental study of visual fixation. In *Psychological Review*, 189.
- Dumais, S.T., and Landauer, T.K., 1984. Describing categories of objects for menu retrieval systems. In *Behavioral Research Methods, Instruments and Computers*, 16(2), 242-248.
- Eades, P.D., 1992. Drawing Free Trees. In *Bulletin of the Institute for Combinatorics and its Applications*, 5, 10-36.
- Ellis, S., and Hitchcock, R., 1986. Emergence of Zipf's Law: Spontaneous Encoding Optimization by Users of a Command Language. In *IEEE Transactions on Systems, Man, and Cybernetics*, 16(3), 423-427.

- Ellis, J., Tran, C., Ryoo, J., and Shneiderman, B., 1995. Buttons vs. menus: An exploratory study of pull-down menu selection as compared to button bars. *Technical Report from the University of Maryland*, 10, 95-106.
- Elm, W.C., and Woods, D.D., 1985. Getting lost: a case study in interface design. In *Proceedings of the Human Factors Society Annual Meeting*, 927-931.
- Engelbart, D.C., 1962. Augmenting human intellect: A conceptual framework. *Summary Report in the Stanford Research Institute*, California.
- Engelbart, D.C., 1968. A Research center for augmenting human intellect. <http://sloan.stanford.edu/MouseSite/1968Demo.html>
- Everett, S.P., and Byrne, M.D., 2004. Unintended effects: varying icon spacing changes users' visual search strategy. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 695-702.
- Feng, J., and Sears, A., 2010. Beyond errors: measuring reliability for error-prone interaction devices. In *Journal Behaviour & Information Technology*, 29(2), 149-163.
- Findlater, L., and McGrenere, J., 2004. A comparison of static, adaptive, and adaptable menus. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 89-96.
- Findlater, L., and McGrenere, J., 2008. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 1247-1256.
- Findlater, L., Moffatt, K., McGrenere, J., and Dawson, J., 2009. Ephemeral adaptation: the use of gradual onset to improve menu selection performance. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 1655-1664.
- Fine, E.M., Woods, R.L., and Peli, E., 2001. Is there a preview benefit when reading scrolled text?. In *Proceedings of Conference in Vision Science and its Applications (VSIA)*.
- Fitts, P.M., and Seeger, C.M., 1953. S-R compatibility: spatial characteristics of stimulus and response codes. In *Journal of Experimental Psychology*, 46, 199-210.

- Fitts, P.M., 1954. The information capacity of the human motor system in controlling the amplitude of movement. In *Journal of Experimental Psychology*, 47(6), 381-391.
- Fitts, P.M., and Peterson, J.R., 1964. Information capacity of discrete motor responses. In *Journal of Experimental Psychology*, 67(2), 103-112.
- Fitts, P.M., and Radford, B.K., 1966. Information capacity of discrete motor responses under different cognitive sets. In *Journal of Experimental Psychology*, 71(4), 475-482.
- Franzke, M., 1995. Turning research into practice: characteristics of display-based interaction. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 421-428.
- Forsey D., 1984. Harmony in Transposition: A Tocatta for Vax and M68000. Master Thesis from the University of Waterloo.
- Fu, W.T., and Gray, W.D., 2004. Resolving the paradox of the active user: Stable suboptimal performance in interactive tasks. In *Cognitive science*, 28, 901-935.
- Furnas, G.W., 1986. Generalized fisheye views. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 16-23.
- Furnas, G.W., Landauer, T.K., Gomez, L.M., and Dumais, S.T., 1987. The vocabulary problem in human-system communication. In *Magazine Communications of the ACM*, 30(1), 964-971.
- Gajos, K.Z., Christianson, D., Hoffmann, R., Shaked, R., Henning, K., Long, J.J., and Weld, D.S., 2005. Fast and robust interface generation for ubiquitous applications. In *Proceedings of International Conference on Ubiquitous Computing (UBICOMP)*, 37-55.
- Gajos, K.Z., Czerwinski, M., Tan, D.S., and Weld, D.S., 2006. Exploring the design space for adaptive graphical user interfaces. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)*, 201-208.
- Gajos, K.Z., Everitt, K., Tan, D.S., Czerwinski, M., and Weld, D.S., 2008. Predictability and accuracy in adaptive user interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 1271-1274.
- Galitz, W.O., 2002. The essential guide to user interface design: an introduction to GUI design principles and techniques, 2<sup>nd</sup> Edition.

- Gan, K.C., and Hoffmann, E.R., 1988. Geometrical conditions for ballistic and visually controlled movements. In *Ergonomics*, 31(5), 829-839.
- Gansner, E.R., and Koren, Y., 2006. Improved Circular Layouts. In *Proceedings of Symposium on Graph Drawing*, 386-398.
- Gaver, W.W., 1991. Technology Affordances. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 79-84.
- Giacomo, E.D., Didimo, W., Liotta, G., and Meijer, H., 2004. Computing Radial Drawings on the Minimum Number of Circles. In *Journal of Graph Algorithms and Applications*, 9(3), 251-261.
- Gibson, J.J., 1977. The theory of affordances. In *Perceiving, acting, and knowing: Toward an ecological psychology*, 67-82.
- Gibson, J.J., 1979. The ecological approach to visual perception.
- Goonetilleke, R., Shih, H.M., On, H.K., and Frisch, J., 2001. Effects of training and representational characteristics in icon design. In *International Journal of Human-Computer Studies*, 54, 741-760.
- Graham, E.D., and MacKenzie, C.L., 1996. Physical versus virtual pointing. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 292-299.
- Gray, W.D., 2000. The nature and processing of errors in interactive behavior. Cognitive. In *Cognitive Science*, 24(2), 205-248.
- Gray, W.D., and Boehm-Davis, D.A., 2000. Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. In *Journal of Experimental Psychology: Applied*. 6(4), 322-335.
- Greenberg, S., and Witten, I., 1993. Supporting Command Reuse. In *International Journal of Man-Machine Studies*, 39(3), 353-390.
- Grossman, T., Kong, N., and Balakrishnan, R., 2007a. Modeling pointing at targets of arbitrary shapes. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 463-472.
- Grossman, T., Dragicevic, P., and Balakrishnan, R., 2007b. Strategies for accelerating on-line learning of hotkeys. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 1591-1600.

- Guiard, Y., Blanch, R., and Beaudouin-Lafon, M., 2004. Object pointing: a complement to bitmap pointing in GUIs. In *Proceedings of Graphics Interface*, 9-16.
- Gutwin, C., and Skopik, A., 2003. Fisheye views are good for large steering tasks. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 201-208.
- Hansen, S.J., Kraut, R.E., and Farber, J., 1984. Interface Design and Multivariate Analysis of Unix commands. In *ACM Transactions on Information Systems (ACM TOIS)*, 2(1), 42-57.
- Halverson, T., and Hornof, A.J., 2004. Explaining eye movements in the visual search of varying density layouts. In *Proceedings of the International Conference on Cognitive Modeling (ICCM)*, 124-129.
- Hartson, H.R., 1998. Human-computer interaction: interdisciplinary roots and trends. In *The Journal of Systems and Software*, 43, 103-118.
- Hemenway, K., 1982. Psychological issues in the use of icons in command menus. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 20-24.
- Hick, W., 1952. On the rate of gain of information. In *Journal of Experimental Psychology*, 4(1), 11-36.
- Hollands, J.G., and Merikle, P.M., 1987. Menu organization and user expertise in information search tasks. In *Journal of the Human Factors and Ergonomics Society*, 29(5), 577-586.
- Hopkins, D., 1987. Direction selection is easy as pie menus. In *Proceedings of Usenix Computer Graphics Workshop*, 12(5), 31-32.
- Hopkins, D., 1991. The design and implementation of pie menus. In *Dr. Dobb's Journal*, 16(12), 16-26.
- Hornof, A.J., and Kieras, D.E., 1997. Cognitive modeling reveals menu search is both random and systematic. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 107-114.
- Hornof, J.A., 2001. Visual search and mouse-pointing in labeled versus unlabeled two-dimensional visual hierarchies. In *ACM Transactions on Computer-Human Interaction (ToCHI)*, 8(3), 171-197.

- Hyman, R., 1953. Stimulus information as a determinant of reaction time. In *Journal of Experimental Psychology*, 45(3), 188-196.
- Inkpen, K.M., 2001. Drag-and-drop versus point-and-click mouse interaction styles for children. In *ACM Transactions on Computer-Human Interaction (ToCHI)*, 8(1), 1-33.
- Jagacinski, R.J., Repperger, D.W., Moran, M.S., Ward, S.L., and Glass, B., 1980. Fitts' law and the microstructure of rapid discrete movements. In *Journal of Experimental Psychology: Human Perception and Performance*, 6(2), 309-320.
- Jankun-Kelly, T.J., and Ma, K.L., 2003. MoireGraphs: Radial focus+context visualization and interaction for graphs with visual nodes. In *Proceedings of IEEE Symposium on Information Visualization (INFOVIS)*, 8.
- Kaptelinin, V., 1993. Item Recognition in menu selection: the effect of practice. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT)* and in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 183-184.
- Keele, S.W., 1973. Attention and human performance.
- Kiger., J.I., 1984. The depth/breadth trade-off in the design of menu-driven user interfaces. In *International Journal of Man-Machine Studies*, 20(2), 201-213.
- Kobayashi, M., and Igarashi, T., 2003. Considering the direction of cursor movement for efficient traversal of cascading menus. In *Proceedings of the annual ACM Symposium on User Interface Software and Technology (UIST)*, 91-94.
- Kopetzky, T., and Mühlhäuser, M., 1999. Visual preview for link traversal on the World Wide Web. In *Proceedings of the International Conference on World Wide Web (WWW)*, 1525-1532.
- Kristensson, P.O., and Zhai, S., 2007. Command strokes with and without preview: using pen gestures on keyboard for command selection. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 1137-1146.
- Kulikoy, S., and Stuerzlinger, W., 2006. Targeted steering motions. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 983-988.
- Kurtenbach, G., 1993. The design and evaluation of marking menus. PhD Thesis from the University of Toronto.

- Kurtenbach, G., and Buxton, W., 1993. The limits of expert performance using hierarchic marking menus. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT)* and in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 482-487.
- Kurtenbach, G., and Buxton, W., 1994. User learning and performance with marking menus. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 258-264.
- Kurtenbach, G., Fitzmaurice, G.W., Owen, R.N., and Baudel, T., 1999. The Hotbox: efficient access to a large number of menu-items. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 231-237.
- Lamping, J., Rao, R., and Pirolli, P., 1995. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 401-408.
- Landauer, T.K., Galotti, K.M., and Hartwell, S., 1983. Natural command names and initial learning: a study of text-editing terms. In *Communications of the ACM*, 26, 495-503.
- Landauer, T., and Nachbar, D., 1985. Selection from alphabetic and numeric menu trees using a touch screen: breadth, depth and width. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 73-78.
- Lane, D.M., Napier, H.A., Batsell, R.R., and Naman, J.L., 1993. Predicting the skilled use of hierarchical menus with the keystroke-level model. In *Human-Computer Interaction (HCI)*, 8(2), 185-192.
- Lane, D.M., Napier, H.A., Peres, S.C., and Sándor, A., 2005. The hidden costs of graphical user interfaces: The failure to make the transition from menus and icons tool bars to keyboard shortcuts. In *International Journal of Human Computer Interaction*, 18(2), 133-144.
- Lazonder, A.W., 1994. Minimalist computer documentation: a study on constructive and corrective skills development. PhD Thesis from the University of Twente.
- Lee, E.S., and Raymond, D.R., 1993. Menu-driven systems. In *Encyclopedia of Microcomputers*, 11, 101-127.

- Liebelt, L.S., McDonald, J.E., Stone, J.D., and Karat, J., 1982. The effect of organization on learning menu access. In *Journal of Human Factors and Ergonomics Society*, 26(6), 546-550.
- Looser, J., Cockburn, A., and Savage, J., 2005. On the validity of using first-person shooters for Fitts' law studies. In *People and Computers, British Computer Society Conference on Human-Computer Interaction*, 2, 33-36.
- Mackay, W.E., 1991. Triggers and barriers to customizing software. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 153-160.
- Mackenzie, I.S., Sellen, A., and Buxton, W., 1991. A comparison of input devices in elemental pointing and dragging tasks. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 161-166.
- MacKenzie, I.S., 1992a. Fitts' law as a research and design tool in human-computer interaction. In *Human-Computer Interaction (HCI)*, 7(1), 91-139.
- MacKenzie, I.S., 1992b. Movement time prediction in human-computer interfaces. In *Proceedings of Graphics Interface (GI)*, 140-150.
- MacKenzie, I.S., Kauppinen, T., and Silfverberg, M., 2001. Accuracy measures for evaluating computer pointing devices. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 9-16.
- MacKenzie, I.S., and Soukoreff, R.W., 2002. Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*, 17, 147-198.
- Maloney, L.T., Trommershäuser, J., and Landy, M.S., 2007. Questions without words: A comparison between decision making under risk and movement planning under risk. In *Integrated Models of Cognitive Systems*, 297-313.
- Mantei, M.M., 1982. Disorientation behavior in person-computer interaction. PhD Thesis from University of Southern California.
- Maxwell, S.E., and Delaney, H.D., 2004. Designing experiments and analyzing data: A model comparison procedure, LEA.
- Mayer, R.E., 1997. From novice to expert. In *Handbook of Human Computer Interaction*, 781-795.
- Mayes, J.T., Draper, S.W., McGregor, A.M., and Oatley, K., 1988. Information flow in a user interface: The effect of experience and context on the recall of MacWrite

- screens. In *Proceedings of Conference of the British Computer Society Human-Computer Interaction (BCS HCI)*, 275-289.
- McDonald, J.E., Stone, J.D., and Liebelt, L.S., 1983. Searching for items in menus: The effects of organization and type of target. In *Proceedings of the Human Factors Society 27th Annual Meeting*, 834-837.
- McGrenere, J., and Ho, W., 2000. Affordances: Clarifying and Evolving a Concept. In *Proceedings of Graphics Interface*, 179-186.
- McGuffin, M., and Balakrishnan, R., 2002. Acquisition of expanding targets. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 57-64.
- McLoone, H., Hinckley, K., and Cutrell, E., 2003. Bimanual interaction on the Microsoft Office Keyboard. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT)*, 49-56.
- Mitchell, J., and Shneiderman, B., 1989. Dynamic versus static menus: an exploratory comparison. In *ACM Special Interest Group on Computer Human Interaction (SIGCHI) Bulletin*, 20(4), 33-37.
- Newell, A., and Card, S.K., 1985. The prospects of psychological science in human-computer interaction. In *Human-Computer Interaction (HCI)*, 1, 209-242.
- Newman, W.M., and Sproull, R.F., 1979. Principles of Interactive Computer Graphics. 2<sup>nd</sup> Edition, 224-225.
- Nielsen, J., 1990. Hypertext and hypermedia. In *Academic Press*.
- Nielsen, J., 1992. Finding usability problems through heuristic evaluation. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 73-380.
- Nielsen, J., 1993. What is usability? In *Usability Engineering*, 36-46.
- Nilsen, E.L., 1991. Perceptual-motor control in human-computer interaction. *Technical Report from the University of Michigan*.
- Norman, D.A., 1988. *The Design of Everyday Things*. New York, Doubleday.
- Norman, D.A., 1999. Affordances, conventions, and design. In *Interactions*, 6(3), 38-41.

- Norman, K. L., Anderson, N. S., Schwartz, J. P., Singer, M. J., Shneiderman, B., Bartol, K., and Weiser, M., 1980. Computer aided decision making and problem solving: A program of research. In *CLC from the University of Maryland*, 19.
- Norman, K.L., and Butler, S., 1989. Search by uncertainty: menu selection by target probability. *Technical Report from the University of Maryland*.
- Norman, K.L., 1991a. Models of mind and machine: Information flow and control between humans and computers. In *Advances in Computers*, 32, 201-254.
- Norman, K.L., 1991b. The psychology of menu selection: designing cognitive control at the human/computer interface.
- Norman, K.L., 2008. *Cyberpsychology: An introduction to the psychology of human-computer interaction*. Cambridge: Cambridge University Press.
- Norman, K.L., and Chin, J.P., 2007. The effect of tree structure on search in a hierarchical menu selection system. In *Journal Behaviour & Information Technology*, 7(1), 51-65.
- Oel, P., Schmidt, P., and Schmitt, A., 2001. Time Prediction of Mouse-based Cursor Movements. In *Proceedings of Conference on Human-Computer Interaction (HCI)*, 2, 37-40.
- O'Malley, C., and Draper, S., 1992. Representation and interaction: Are mental models all in the mind? In *Models in the mind: Theory, perspectives and applications*, 73-92.
- Paap, K.R., and Roske-Hofstrand, R.J., 1986. The optimal number of menu options per panel. In *Journal Human Factors & Ergonomics Society*, 28(4), 377-385.
- Padilla, M., 2003. Strike a balance: Users' expertise on interface design. Understanding Web application architecture and the user experience aids in interface design decisions. <http://www.ibm.com/developerworks/web/library/wa-ui/>
- Park, J., and Kim, J., 2000. Effects of contextual navigation aids on browsing diverse Web systems. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, USA, 257-264.
- Parkinson, S.R., Hill, M.D., Sisson, N., and Viera, C., 1988. Effects of breadth, depth and number responses on computer menu search performance. In *International Journal of Man-Machine Studies*, 28(6), 683-692.

- Pastel, R.L., 2006. Measuring the Difficulty of Steering Through Corners. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 1087-1096.
- Payne, S.J., 1991. Display-based action at the user interface. In *International Journal of Man-Machine Studies*, 35(3), 275-289.
- Peres, S.C., Tamborello, F.P., EE, Fleetwood, M.D., Chung, P., and PaigeSmith, D.L., 2004. Keyboard shortcut usage: The roles of social factors and computer experience. In *Proceedings of the Human Factors and Ergonomics Society*, 803-807.
- Peres, S.C., Fleetwood, M.D., Tamborello, F.P., II, Yang, M., 2005. The pros and cons of efficiency: Predicting the use of the keyboard to issue commands. In *Proceedings of the Human Factors and Ergonomics Society*, 637-641.
- Phillips, J.G., and Triggs, T.J., 2001. Characteristics of cursor trajectories controlled by the computer mouse. In *Ergonomics*, 44(5), 527-536.
- Pirolli, P.L.T. Information foraging theory: adaptive interaction with information.
- Pirolli, P.L.T., and Card, S.K., 1999. Information foraging. In *Psychological Review*, 106(4), 643-675.
- Pirolli, P.L.T., Card, S.K., and Van Der Wege, M., 2003. The effects of information scent on visual search in the hyperbolic tree browser. In *ACM Transactions on Computer-Human Interaction (ToCHI)*, 10(1), 20-53.
- Pomerantz, J.R., 1981. Perceptual Organization in Information Processing. In *Perceptual Organization*, 141-180.
- Preece, J., and Rogers, Y., 1994. Human-computer interaction. Addison-Wesley Publishing Company.
- Prümper, J., Frese, M., Zapf, D., and Brodbeck, F.C., 1991. Errors in computerized office work: differences between novice and expert users. In *ACM Special Interest Group on Computer Human Interaction (SIGCHI) Bulletin*, 23(2), 63-66.
- Quinn, P., and Cockburn., A., 2008. The effects of menu parallelism on visual search and selection. In *Proceedings of Conference on Australasian User Interface*, 76, 79-84.

- Raymond, D.R., 1986. A survey of research in computer-based menus. *Research Report from the University of Waterloo (Ontario)*.
- Rayner, K., 1978. Eye movements in reading and information processing. In *Psychological Bulletin*, 85, 618-660.
- Rieman, J., 1996. A field study of exploratory learning strategies. In *ACM Transactions on Computer-Human Interaction (ToCHI)*, 3(3), 189-218.
- Robertson, G., McCracken, D., and Newell, A., 1981. The ZOG approach to man-machine communication. In *International Journal of Man-Machine Studies*, 14, 461-488.
- Rock, I., and Palmer, S., 1990. The legacy of Gestalt psychology. In *Scientific American*, 263(6), 84-90.
- Roudaut, A., Bailly, G., Lecolinet, E., and Nigay, L., 2009. Leaf Menus: Linear menus with stroke shortcuts for small handheld devices. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT)*, 616-619.
- Samp, K., and Decker, S., 2010. Supporting menu design with radial layouts. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI)*, 155-162.
- Samp, K., and Decker, S., 2011a. Navigation Time Variability: Measuring Menu Navigation Errors. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT)*.
- Samp, K., and Decker, S., 2011b. Visual Search in Radial Menus. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT)*.
- Schmidt, R.A., Zelaznik, H., Hawkins, B., Frank, J.S., and Quinn, J.T., 1979. Motor-output variability: A theory for the accuracy of rapid motor acts. In *Psychological Review*, 86, 415-451.
- Schultz, E.Jr., and Curran, P.S., 1986. Menu structure and ordering of menu selections: Independent or interactive effects? In *ACM Special Interest Group on Computer Human Interaction (SIGCHI) Bulletin*, 18(2), 69-71.
- Sears, A., and Shneiderman, B., 1994. Split menus: effectively using selection frequency to organize menus. In *ACM Transactions on Computer-Human Interaction (ToCHI)*, 1(1), 27-51.

- Seppala, P., and Salvendy, G., 1985. Impact of depth of menu hierarchy on performance effectiveness in a supervisory task: Computerized flexible manufacturing system. In *Human Factors*, 27, 713-722.
- Sheikh, I.H., and Hoffmann, E.R., 1994. Effect of target shape on movement time in a Fitts' task. In *Ergonomics*, 37, 1533-1547.
- Shelton, D., and Newhouse, R., 1981. Incidental learning in a paired-associate task. In *Journal of Experimental Education*, 50, 36-38.
- Sheridan, M.R., 1979. A reappraisal of Fitts' law. In *Journal of Motor Behavior*, 11, 179-188.
- Shneiderman, B. 1983. Direct manipulation: a step beyond programming languages. In *IEEE Computer*, 16(8), 57-69.
- Shneiderman, B., and Plaisant, C., 2004. Designing the User Interface (4<sup>th</sup> Edition).
- Six, J.M., and Tollis, I.G., 1999. A Framework for Circular Drawings of Networks. In *Proceedings of Symposium on Graph Drawing*, 107-116.
- Smyrnis, N., Evdokimidis, I., Constantinidis, T.S., and Kastrinakis, G., 2000. Speed-Accuracy trade-off in the performance of pointing movements in different directions in two dimensional-space. In *Experimental Brain Research*, 134 (1), 21-31.
- Snowberry, K., Parkinson, S. R., and Sisson, N., 1983. Computer display menus. In *Ergonomics*, 26(7), 699-712.
- Somberg, B.L., 1987. A comparison of rule-based and positionally constant arrangements of computer menu items. In *Proceedings of the Conference on Human Factors in Computing Systems and Graphics Interface (CHI/GI)*, 255-260.
- Soukoreff, R.W., and MacKenzie, I.S., 2004. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International Journal Human-Computer Studies*, 61(6), 751-789.
- Tamborello, F.P., II, Peres, S.C., and Fleetwood, M.D., 2006. A peer-mediated leap to efficiency: A theory of cost-benefit analysis in the selection of efficient strategies. In *Proceedings of the Human Factors and Ergonomics Society*, 349-353.

- Tanvir, E., Cullen, J., Irani, P., and Cockburn, A., 2008. AAMU: adaptive activation area menus for improving selection in cascading pull-down menus. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 1381-1384.
- Tauscher, L., and Greenberg, S., 1997. How people revisit web pages: empirical findings and implications for the design of history systems. In *International Journal of Human Computer Studies*, 47, 97-137.
- Temple, Barker, and Sloane, Inc., 1990. The benefits of the graphical user interface.
- Thompson, S.G., 2007. Effect on movement performance as a function of visual-motor scale and velocity: An investigation of the speed-accuracy tradeoff. PhD Thesis from the University of Louisiana at Monroe.
- Treisman, A., 1982. Perceptual grouping and attention in visual search for features and for objects. In *Journal of Experimental Psychology: Human Perception Performance*, 8(2), 194-214.
- Trommershäuser, J., Gepshtein, S., Maloney, L.T., Landy, M.S., and Banks, M.S., 2005. Optimal compensation for changes in effective movement variability. In *Journal of Neuroscience*, 25, 7169-7178.
- Trommershäuser, J., Maloney, L.T., and Landy M.S., 2008. The expected utility of movement. In *Neuroeconomics*, 95-111.
- Walker, N., Meyer, D.E., and Smelcer, J.B., 1993. Spatial and temporal characteristics of rapid cursor-positioning movements with electromechanical mice in human-computer interaction. In *Human Factors*, 35(3), 431-458.
- Wallace, R.J., 1971. S-R compatibility and the idea of a response code. In *Journal of Experimental Psychology*, 88, 354-360.
- Welford, A.T., 1968. Fundamentals of skill.
- Welford, A.T., Norris, A.H., and Shock, N.W., 1969. Speed and accuracy of movement and their changes with age. In *Acta Psychologica*, 30, 3-15.
- Wendel, R., and Frese, M., 1987. Developing exploratory strategies in training: The general approach and a specific example for manual use. In *Proceedings of the Conference on Human-Computer Interaction (INTERACT)*, 943-948.
- Wertheimer, M., 1923/1958. Principles of Perceptual Organization. In *Readings in Perception*, 115-135.

- Whalen, T., and Mason, C., 1981. The use of tree-structured index which contains three types of design defects in the design of videotext tree indexes. In *Behavioral Research and Evaluation*, 15-34.
- Whisenand, T.G., and Emurian, H.H., 1997. Analysis of cursor movements with a mouse. In *Proceedings of the Conference on Human-Computer Interaction (HCI)*, 1, 533-536.
- Whittaker, S., Terveen, L., and Nardi, B.A., 2000. Let's stop pushing the envelope and start addressing it: a reference task agenda for HCI. In *Human-Computer Interaction*, 15, 75-106.
- Wiseman, N.E., Lemke, H.U., and Hiles, J.O., 1969. PIXIE: A New Approach to Graphical Man-machine Communication. In *Proceedings of CAD Conference Southhampton*, 463.
- Woodruff, A., Faulring, A., Rosenholtz, R., Morrision, J., and Pirolli, P., 2001. Using thumbnails to search the Web. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 198-205.
- Worden, A., Walker, N., Bharat, K., and Hudson., S., 1997. Making computers easier for older adults to use: area cursors and sticky icons. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 266-271.
- Wright, C.E., and Meyer, D.E., 1983. Conditions for a linear speed-accuracy trade-off in aimed movements. In *Journal of Experimental Psychology: Human Experimental Psychology*, 35A(2), 279-296.
- Wright, P., 1988. Communicating with the user. In *Designing end-user interfaces*, Pergamon State of the Art Report, 15(8), 121-129.
- Yee, K.P., Fisher, D., Dhamija, R., Hearst, M., 2001. Animated Exploration of Dynamic Graphs with Radial Layout. In *IEEE Symposium on Information Visualization (INFOVIS)*, 43-50.
- Yu, B.M., and Roh, S.Z., 2002. The effects of menu design on information-seeking performance and user's attitude on the World Wide Web. In *Journal of the American Society for Information Science and Technology (JASIST)*, 53(11), 923-933.
- Zelaznik, H.N., Mone, S., McCabe, G.P., and Thaman, C., 1988. Role of temporal and spatial precision in determining the nature of the speed-accuracy trade-off in

- aimed-hand movements. In *Journal of Experimental Psychology: Human Perception and Performance*, 14(2), 221–230
- Zhai, S., Conversy, S., Beaudouin-Lafon, M., and Guiard, Y., 2003. Human on-line response to target expansion. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 177-184.
- Zhai, S., 2004. Characterizing computer input with Fitts' law parameters: the information and non-information aspects of pointing. In *International Journal Human-Computer Studies*, 61(6), 791-809.
- Zhao, S., and Balakrishnan, R. 2004. Simple vs. compound mark hierarchical marking menus. In *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, 33-42.
- Zhao, S., Agrawala, M., and Hinckley., K., 2006. Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 1077-1086.
- Zhou, X., Cao, X., and Ren, X., 2009. Speed-accuracy tradeoff in trajectory-based tasks with temporal constraint. In *Proceedings of the International Conference on Human-Computer Interaction (INTERACT)*, 906-919.
- Zhou, X., and Ren, X., 2010. An investigation of subjective operational biases in steering tasks evaluation. In *Journal Behaviour & Information Technology*, 29(2), 125-135.