

# The Usability of CommandMaps in Realistic Tasks

Joey Scarr<sup>†</sup>, Andy Cockburn<sup>†</sup>, Carl Gutwin<sup>‡</sup>, Andrea Bunt<sup>\*</sup>, Jared Cechanowicz<sup>\*</sup>

<sup>†</sup>Computer Science

University of Canterbury  
Christchurch, New Zealand

<sup>‡</sup>Computer Science

University of Saskatchewan  
Saskatoon, Canada

<sup>\*</sup>Computer Science

University of Manitoba  
Winnipeg, Canada

{joey,andy}@cosc.canterbury.ac.nz, gutwin@cs.usask.ca, bunt@cs.umanitoba.ca

## ABSTRACT

CommandMaps are a promising interface technique that flattens command hierarchies and exploits human spatial memory to provide rapid access to commands. CommandMaps have performed favorably in constrained cued-selection studies, but have not yet been tested in the context of real tasks. In this paper we present two real-world implementations of CommandMaps: one for Microsoft Word and one for an image editing program called Pinta. We use these as our experimental platforms in two studies. In the first, we show that CommandMaps demonstrate performance and subjective advantages in a realistic task. In the second, we observe naturalistic use of CommandMaps over the course of a week, and gather qualitative data from interviews, questionnaires, and conversations. Our results provide substantial insight into users' reactions to CommandMaps, showing that they are positively received by users and allowing us to provide concrete recommendations to designers regarding when and how they should be implemented in real applications.

## Author Keywords

CommandMaps; spatial memory; hierarchies; real tasks.

## ACM Classification Keywords

H.5.2 [User Interfaces]: Interaction Styles.

## INTRODUCTION

The CommandMap, introduced by Scarr et al. [23], is a full-screen interface intended to replace hierarchical menus and toolbars (Figure 1). When displayed by pressing a command key or clicking a dedicated button, a CommandMap uses the whole screen to display functionality, flattening the command hierarchy and exploiting users' strong ability to remember item locations in spatial layouts. These properties theoretically enable expert CommandMap users to reach a high level of performance. Furthermore, since they are based on visual, point-and-click interaction, they require little training and

allow users to maintain a predominantly 'see and point versus learn and remember' modality of interaction that assists novice users [24]. However, there are also risks to the CommandMap approach: for example, the complexity of the full-screen UI may deter novice users, and the workspace occlusion caused by the CommandMap's modal presentation may disrupt the user's workflow.

Scarr et al. validated the CommandMap design in a series of command selection experiments. In brief, their results demonstrated that once users were familiar with the CommandMap layout and interaction mechanics, they could access commands more quickly than when using traditional menus or the Microsoft Office Ribbon user interface. These are promising and important findings, showing that CommandMaps have the *potential* to improve performance in the frequent and basic task of selecting commands. However, there are many factors influencing UI success in the real world that were not present in these lab studies; consequently, there are several further research questions that need to be answered before CommandMaps can be deployed with confidence in real-world software.

The fundamental question motivating this research is "*Do CommandMaps work in the real world?*" From a methodological perspective, this is a critical question to answer, especially when proposing an interface mechanism that changes the basic method for presenting and accessing commands. The importance of addressing this question is accentuated by the many previous HCI studies that have demonstrated divergence between results obtained in the lab and in the field, with lab studies typically generating stronger, more positive findings [5, 8, 20]. Further, the importance of triangulating results through complementary methods is well known in the behavioral and social sciences [17], including HCI [14].

More specifically, this paper comprises an attempt to answer the following research questions:

**Q1) Do CommandMaps demonstrate performance and subjective advantages in real tasks?** Real tasks, such as document formatting or image editing, have many characteristics that were not present in previous studies. For example, an individual command selection usually forms a small step of a larger task, which may also include content creation activities such as typing or drawing. If CommandMaps somehow interfere with the other cognitive and physical activities involved in task completion, then these drawbacks may outweigh the benefits.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2473-1/14/04...\$15.00.

<http://dx.doi.org/10.1145/2556288.2556976>

**Q2)** *How do users respond when first using CommandMaps?* Scarr et al.'s previous studies showed that users without item location knowledge were not adversely affected by CommandMaps, but participants were allowed time to become familiar with the interface first. Therefore, the issue of users' initial 'out of the box' experience, both in terms of performance and subjective impressions, requires further examination.

**Q3)** *How do performance and subjective impressions change in real tasks as users gain experience?* While previous studies have shown that performance and subjective preferences favor CommandMaps after condensed periods of command selection, the progression from novice to expert over a realistic period of time (e.g., a week) has not been studied.

**Q4)** *Do CommandMaps generalize to multiple application contexts?* In order to recommend the use of CommandMaps in the real world, we need to know whether they are successful in different application genres (such as word processing and image editing), and whether there are characteristics of different applications that make CommandMaps more or less useful.

**Q5)** *What usability issues arise with CommandMaps in real-world use?* By observing real use, we can identify usability issues and improve their design.

In order to answer these questions, we examine the use of CommandMaps in realistic tasks. We first present two implementations that modify real-world software systems to use CommandMaps – Figure 1 shows these systems, based on Microsoft Word and the open-source painting and photo editing application 'Pinta'. These implementations serve as our experimental platforms in two studies.

The first study, addressing **Q1**, examines how users complete moderately realistic word processing tasks when using the Microsoft Word CommandMap. For baseline comparison, we also examined an equivalent task when using an unaltered version of Microsoft Word 2007. Results show that the CommandMap was at least as fast or faster than the unaltered Ribbon UI (depending on the amount of tab switching required), and was strongly preferred.

The second study, addressing **Q2-5**, focuses on user opinions, and the evolution of those opinions, during the first week of interaction with CommandMaps. Participants interacted with a CommandMap and the standard interface in both Microsoft Word and Pinta. They were given realistic tasks to complete; they were invited to complete them in any way they saw fit; and they repeated the tasks every day for a week. Observations of use, interviews, and questionnaires gave us a rich pool of data, from which we determined the following key results: 1) many participants, even those highly familiar with the standard Word interface, preferred the CommandMap after using it for a week; 2) novice users found the CommandMap useful for gaining an overview of the interface; 3) performance increased at about the same rate as standard interfaces; 4)

CommandMaps were most effective in replacing interfaces with deep menu hierarchies; and 5) there are several specific opportunities to improve CommandMaps' usability (described later in the paper).

This work makes three primary contributions. First, we present two real-world implementations of CommandMaps in very different application domains (word processing and graphics editing). Second, our studies provide clear and compelling evidence that CommandMaps are a usable and learnable interface alternative that provides higher performance, reduced search times, and higher approval ratings from most users. Our studies, combined with the laboratory work that has been done previously, strongly indicates that UI designers should adopt the CommandMap idea and its foundational principles of flattened command hierarchies, spatial memory, and fast access. Finally, the results of our second study allow us to recommend design alterations to CommandMaps, as well as providing concrete advice regarding when and how designers should consider implementing them in real-world applications.

## RELATED WORK

Building a credible body of empirical evidence in support of the CommandMap design is a critical motivator for this work. In our treatment of previous work, therefore, we first consider the importance of validating an experimental hypothesis using multiple research methods. Second, we perform a review of systems similar to CommandMaps and discuss the experimental evidence available for each.

### Experimental Triangulation

In the design of any research question, there exists a tension between the goals of internal and external validity. Mackenzie [15] discusses this trade-off, noting that strong empirical results often require a narrow research question (e.g., "does Technique A provide faster selection time than Technique B after one hour of use?"), while striving for external validity leads to broad, less easily testable questions ("is Technique A better than Technique B?"). While the former approach produces replicable empirical data, it can often fail to answer the broader question of whether a technique will be successful in the real world. McGrath [17] asserts that "it is only by accumulating evidence, over studies done so that they involve different – complementary – methodological strengths and weaknesses that we can begin to consider the evidence as credible, as probably true, as a body of empirically-based knowledge."

Mackay and Fayard [14] concur that "it is simply not possible to use a single study that addresses all the possible threats to validity", and argue that the solution is triangulation, or "using more than one research approach to address the same question" (p. 231). The importance of using more than one research approach to answer the same question is highlighted by several case studies in HCI (see [5, 8, 20]). In these examples, initial laboratory studies provided positive results, but later field studies revealed usability issues only present in real-world conditions.

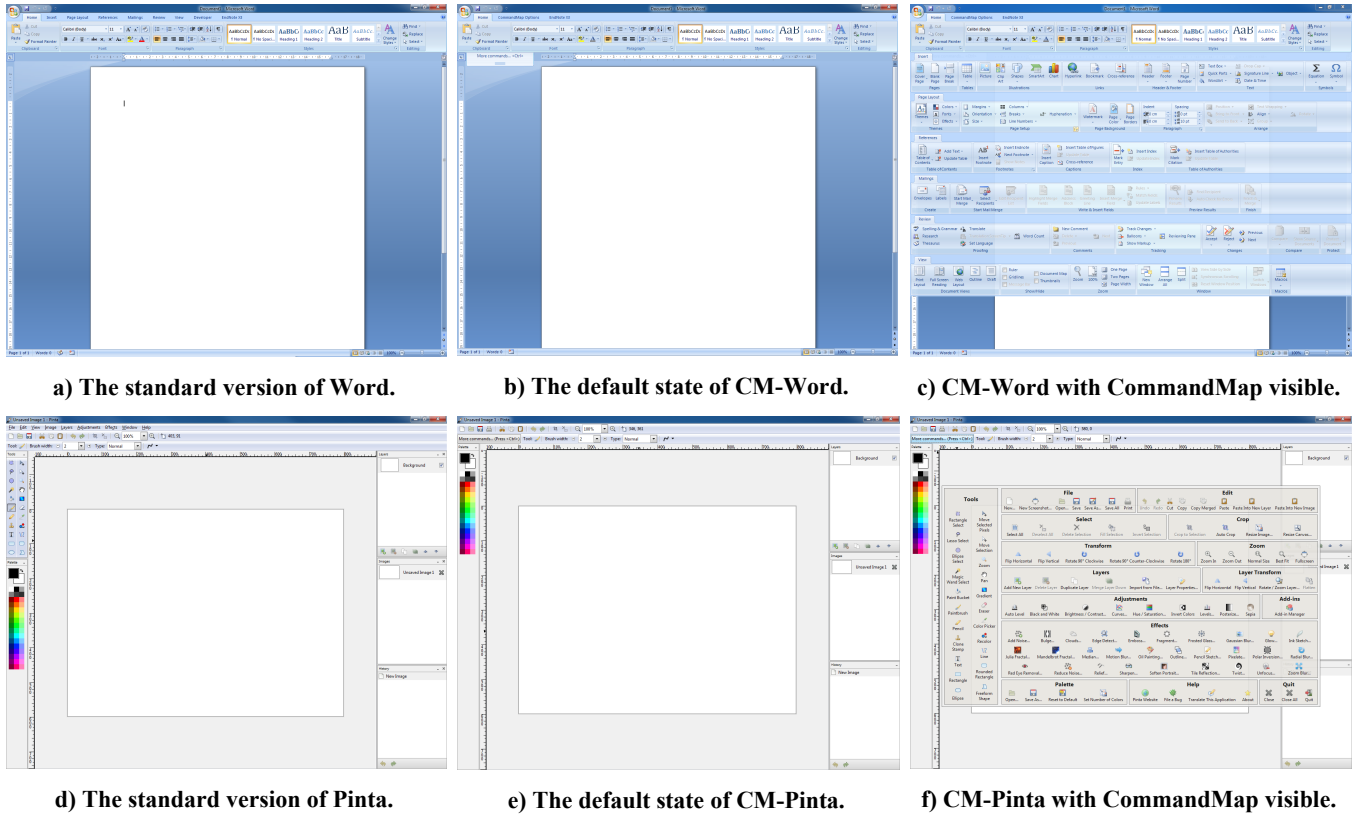


Figure 1. Standard (left) and CommandMap (middle, right) versions of Word and Pinta.

### Parallel Menus and Full-Screen Command Layouts

One way of supporting rapid, point-and-click selection of items in a graphical user interface is to present items in a flat, spatially stable structure. Interfaces such as these support rapid selection by leveraging two complementary facets of performance. The first is human memory for item locations, which develops automatically as a by-product of interaction [11] and facilitates extremely quick item revisitation over time [6, 19, 23]. We refer the reader to Scarr et al. [22] for a review of spatial memory. The second is that when item locations are known, flat hierarchies allow optimal retrieval time (i.e., flat structures allow users to point directly to the target, as opposed to having to navigate through a hierarchical control structure) [2]. Spatially stable, full-screen 2D layouts have been investigated in several application domains, including bookmark browsing [21] and document navigation [3], with results showing strong performance benefits for revisitation in each case.

Scarr et al. [23] presented the CommandMap as a spatially-stable interaction technique that used flat hierarchies to provide quick access to commands. They validated their prototype system using highly repetitive command selection tasks, showing that expert users were able to perform faster command selections compared to menus and the Microsoft Office Ribbon. Gutwin and Cockburn’s earlier ListMap design [6] showed similar advantages for experts, but reduced performance for novices (which they attributed to visual search difficulties in the constrained interface space).

Selection time benefits of parallel command menus have also been demonstrated by Hochheiser and Shneiderman [9], whose experiment showed that simultaneous menus on websites provided advantages for “experienced users performing complex tasks” compared to hierarchical menus. Malacria et al. [16] proposed a related method using menu parallelization to display hotkey bindings in their ExposeHK system, although it was not evaluated.

To date, research on parallelized command menus such as CommandMaps is promising but limited. In particular, while the previous CommandMap studies provide compelling evidence in favor of their use, the studies were performed with cued-selection tasks, which omit many of the subtleties and usability issues involved with completing tasks in the real world.

### A COMMANDMAP FOR MICROSOFT WORD

Our implementation of the Microsoft Word CommandMap is based on Scarr et al.’s original prototype, with some adjustments (Figure 1b,c). The prototype displayed no commands until a command key was pressed, at which point all of the commands were displayed in parallel, overlaying the document. We modified this design for Microsoft Word CommandMap so that the *Home* tab was continually displayed at the top of the window (see Figure 1b). We had two main reasons for this: first, it provides convenient visual access to the most frequently used commands; and second, it provides a visual focus of attention for displaying a tip for novice CommandMap

users that instructs them on how to access other commands. This tip takes the form of a dropdown tab with the label “More commands... <Ctrl>” (see top-left of Figure 1b). When the “More commands” tab is clicked, or when the <Ctrl> key is held, the CommandMap is displayed, making all commands from all tabs available at once by pointing and clicking (see Figure 1c). Releasing the <Ctrl> key without making a selection cancels the command and hides the CommandMap. Selecting a command completes the command and hides the CommandMap.

### Implementation and Design

The CommandMap is implemented as a plug-in for Microsoft Word 2007. To assist replication or further development, the plug-in is available at [www.cosc.canterbury.ac.nz/research/RG/HCI/HCIlab/](http://www.cosc.canterbury.ac.nz/research/RG/HCI/HCIlab/). The plug-in uses an open-source, third-party Ribbon control ([officeribbon.codeplex.com/](http://officeribbon.codeplex.com/)) to mimic the look and feel of the original Word 2007 Ribbon. The visual appearance of controls is replicated using the original Word 2007 icons. When the CommandMap is displayed, either by holding <Ctrl> or by clicking the on-screen drop-down button, the CommandMap appears as a borderless overlay at 95% opacity on top of the main Word window. It displays tabs in a vertical arrangement below the already-visible Home tab. Interaction with the controls in the CommandMap works the same way as the Ribbon: clicking an item invokes its corresponding command using the Word Object Model.

Although the plug-in provides good coverage of Microsoft Word’s interface capabilities in a fluid and convincing user interface, certain features of Word were difficult or impossible to replicate using the plug-in approach. For example, contextual tabs, which are only visible in certain editing contexts (such as when a table or image is selected), still appear in the topmost Ribbon, as do the tabs of other Word plug-ins, such as EndNote. “Galleries” of pre-defined content (such as the WordArt, Symbol and Equation menus) were also excluded, due to the significant amount of engineering required to replicate them.

### A COMMANDMAP FOR PINTA

To explore the application and use of CommandMaps in a domain other than word processing, we also designed and implemented a CommandMap for an image editing application. We chose an image editing application because content creation and editing of images is predominantly mouse-based, which is substantially different to the keyboard creation and editing actions when word processing. We selected Pinta, a version of Paint.NET, as the base application because its C# codebase is open source. The adapted version of Pinta replaces its menus with a CommandMap, requiring various interface adaptations to the CommandMap concept, as described below.

### Implementation and Design

Pinta is implemented in C#, using the .NET Framework. We forked the code and built a CommandMap-based version, as shown in Figure 1d-f.

Image editing programs such as Pinta typically have toolbars, as well as various palettes (such as Tools, Colors, Layers, and History) that are docked around the edge of the screen. This raises the design question: *when converting an application’s interface to a CommandMap, what subset of controls should the CommandMap contain?*

User interface palettes are often used to provide convenient ready-to-hand access to related controls, thus obviating the need for the user to navigate through menu structures for those controls. Palettes, therefore, serve a related function to CommandMaps, although for much smaller command subsets. Therefore, in our version of Pinta, we left the always-visible palettes and toolbars as-is, and included only the menu-based commands in our CommandMap. One exception was the Tools palette, which we moved into the CommandMap to encourage our study participants to use the CommandMap more often.

### STUDY ONE

In this study, our goal was to answer **Q1**: *do CommandMaps demonstrate performance and subjective advantages in real tasks?* We therefore compared user performance and subjective assessments for a word processing task when executed using the unmodified version of Microsoft Word and when using it with our CommandMap plug-in installed. Participants were given written instructions describing the formatting of a document. The tasks involved elements of text selection, command execution and typing. Each participant completed the tasks using both versions of Microsoft Word.

### Participants and Apparatus

Ten paid participants (7 male, 3 female; aged 18-34 years) were recruited from a local university. All were familiar with Microsoft Word 2007, using it for a mean of 15.6 hours/week, and none had seen a CommandMap UI before.

The experiment was performed on a quad-core Windows 7 machine with Microsoft Word 2007 installed. Participants used a 23” monitor running at 1920×1080 resolution, and a standard keyboard and mouse.

### Procedure

Participants first completed a demographics questionnaire. They then completed two experimental tasks (‘Task One’ and ‘Task Two’, see Table 1), with one performed using the CommandMap and the other using the standard Word Ribbon. The order of CommandMap and standard Ribbon interfaces was counterbalanced between participants. To familiarize the participant with both the upcoming interface and the commands used in the upcoming task, a brief practice task (not timed) was performed immediately prior to both Task One and Task Two. Participants timed themselves by clicking on-screen Start and Stop buttons before and after each task, and were instructed to complete tasks as quickly and accurately as possible.

In order to measure initial impressions of the CommandMap and analyze whether opinions changed after

actual use, we administered a Likert-scale questionnaire (Table 2) immediately after participants were first shown the CommandMap, as well as an equivalent one after participants had completed a task with it. Participants also completed an unweighted NASA-TLX [7] survey after using each interface (Table 3) and gave written comments.

### Tasks

With each interface, participants performed a practice task followed by a timed task. The nature of the task and the instructions were nearly identical between the practice task and the following experimental task, with the only difference being that the practice task was performed on a smaller document requiring fewer changes. Tasks One and Two were always performed in the order One, then Two (although, to repeat, the interface order was counterbalanced between them). Each task involved 25 change review subtasks, followed by 14 annotation subtasks (see Table 1 for a summary).

Task One operations	Task Two operations
Select text Click <i>Accept Change</i> (automatically moves to next change) Go back and re-select accepted text Click <i>Strikethrough</i>	Select text Click <i>Reject Change</i> (automatically moves to next change) Go back and re-select rejected text Click <i>Superscript</i>
25 repetitions	25 repetitions
Select paragraph Click <i>Word Count</i> Memorize number of words Close <i>Word Count</i> dialog Click <i>Insert Footnote</i> Type word count into footnote	Select paragraph Click <i>Word Count</i> Memorize number of words Close <i>Word Count</i> dialog Click <i>Insert Comment</i> Type word count into comment
14 repetitions	14 repetitions

**Table 1. The operations required for each task in Study One.**

To minimize learning effects, the tasks were designed to be isomorphic, but not identical (Word Count was the only command used in both). As Study One is primarily concerned with performance issues and initial impressions, the experimental tasks are designed to focus on the key difference between the user interfaces – their methods of command selection. We note, however, that command-heavy usage is not always unrealistic: for example, Lafreniere et al.’s study of application use [13] showed single sessions that contained up to 1489 separate command invocations. Study Two, reported later, examines less constrained experimental tasks.

### Design

Our experimental analysis was structured as a 2×2 mixed design ANOVA, with within-subjects factor *Interface* (*standard*, *commandmap*) and between-subjects factor *Task Pairing* (*cm-first*, *standard-first*). We included Task Pairing as a factor in our analysis to check for any asymmetric skill transfer effects, which might stem from one of the interfaces in Task 1 serving as better preparation for the second interface in Task 2. Our primary dependent variable was task completion time; we also analyzed TLX responses for each condition, and responses to the Likert-scale questionnaires before and after use of the CommandMap.

### Hypotheses

Previous studies of CommandMaps examined command selection time in response to a stimulus that identified the target command. In this experiment, the stimuli for command selections are placed into a more realistic interaction context, where selections are made to enact actual changes to the document. This difference could potentially influence experimental results for CommandMaps, for at least two reasons. First, in making a series of changes to a document, participants are likely to be expending some of their cognitive resources on activities such as remembering their location within the document – this may decrease the advantage of using a CommandMap, which relies on spatial memory. Second, using the CommandMap involves a substantial display context switch, with the CommandMap temporarily obscuring the work surface. This may interrupt the user’s higher-level task flow, including their memory of their location in the document, how far through the task they were, or what they were supposed to do next. However, the same risks apply to standard user interfaces – for example, a user searching through tabs in the Ribbon to find a command may be just as likely to lose task flow. We therefore hypothesized that the previously observed benefits for CommandMaps would continue to hold. Our formal hypotheses were:

- H1.** Subjects will complete tasks more quickly with the CommandMap than with the standard interface.
- H2.** Subjects will prefer CommandMap to the standard UI.

In addition, we planned an exploratory comparison between the Likert-scale questionnaire measures before and after CommandMap use (Table 2), in order to quantify whether users’ initial impressions of CommandMaps changed.

### Results

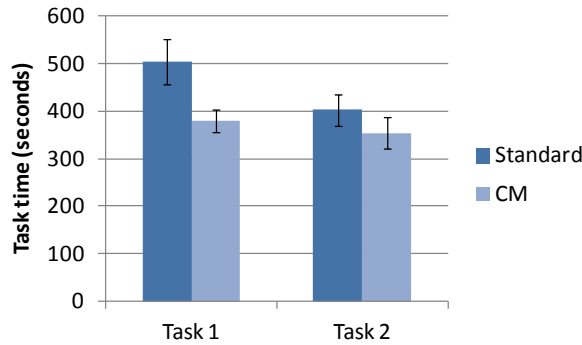
#### Task time

A two-way ANOVA on task time showed a significant main effect of *Interface* ( $F_{1,8} = 31.6$ ,  $p < 0.001$ ,  $\eta^2 = 0.8$ ), with *commandmap* (mean 366s) significantly faster than *standard* (mean 453s). There was also a significant interaction effect between *Interface* and *Task Pairing* ( $F_{1,8} = 17.0$ ,  $p = 0.03$ ). Figure 2 depicts the nature of this interaction effect, suggesting that while *commandmap* was faster overall, the benefit was more pronounced for Task 1 than Task 2. Participant comments suggested that Task 2 may have been easier with the Ribbon interface than Task 1, with one noting that Task 2 required fewer tab switches (the sequence [Word Count, Insert Comment] requires a tab switch while [Word Count, Footnote] does not, leading to 78 tab switches in Task One compared to 50 in Task Two). Previous research [23] has also shown that user performance with the Ribbon is highly dependent on the number of tab switches involved. Examining performance on each task individually using between-subjects analyses, we find a significant effect of *Interface* in Task 1 ( $F_{1,8} = 5.5$ ,  $p = 0.046$ ), but none in Task 2 ( $F_{1,8} = 1.1$ ,  $p = 0.3$ ). We therefore partially accept **H1**. An alternative explanation of

Question (1 = Disagree, 5 = Agree)	Before CM use [mean (s.d.)]	After CM use [mean (s.d.)]	Wilcoxon z	Sig (p)
Q: It will be/was easy to find controls in the full-screen interface.	3.8 (0.6)	4.0 (0.7)	0.7	0.2
<b>Q: I will be/was able to remember the locations of controls in the full-screen interface.</b>	<b>3.5 (0.9)</b>	<b>4.4 (0.7)</b>	<b>2.3</b>	<b>0.01</b>
Q: The full-screen interface is/was visually overwhelming.	3.2 (0.9)	2.9 (1.0)	0.8	0.2
<b>Q: I will be/was able to interact quickly with the full-screen interface.</b>	<b>3.5 (1.0)</b>	<b>4.4 (0.8)</b>	<b>2.5</b>	<b>0.01</b>
Q: I would be discouraged from using software with this kind of interface in the future.	2.0 (0.9)	1.6 (0.8)	1.6	0.05

**Table 2. Likert-scale questions and responses before and after CommandMap use in Study One. Significant differences in bold.**

the interaction is that asymmetric skill transfer occurred – although it might be argued that the CommandMap was particularly good preparation for the Normal user interface in Task 2 (explaining the larger improvement across Tasks One and Two with Normal than with CommandMaps), the previous explanation seems more compelling. Further, even if the explanation of asymmetric skill transfer is correct, then this would suggest a further and unexpected benefit of CommandMaps in helping users learn standard interfaces.



**Figure 2. Mean task times in Study One. Error bars  $\pm 1$  s.e.m.**

#### Subjective responses

NASA-TLX responses (Table 3) showed that participants found CommandMaps to be less physically demanding, less temporally demanding, easier to use and less frustrating than the standard interface. Subject preferences were also in favour of CommandMaps, with nine out of ten users preferring CommandMaps and one preferring the standard interface ( $\chi^2 = 4.9$ ,  $p = 0.03$ ); we therefore accept **H2**.

	Standard	CommandMap	Wilcoxon z	Sig
Mental demand	2.8 (0.9)	2.6 (1.0)	0.4	0.34
<b>Physical demand</b>	<b>3.2 (1.1)</b>	<b>1.9 (0.9)</b>	<b>2.2</b>	<b>0.01</b>
<b>Temporal demand</b>	<b>3.1 (1.1)</b>	<b>2.2 (0.6)</b>	<b>1.9</b>	<b>0.03</b>
Success	4.2 (0.4)	4.3 (0.5)	0.0	0.5
<b>Hard work</b>	<b>3.2 (1.2)</b>	<b>2.1 (0.9)</b>	<b>1.9</b>	<b>0.03</b>
<b>Frustration</b>	<b>2.5 (1.4)</b>	<b>1.6 (0.8)</b>	<b>1.8</b>	<b>0.04</b>

**Table 3. Mean (st. dev.) NASA-TLX values (1=low, 5=high).**

**Rows in bold indicate significant differences.**

Results for the CommandMap questionnaire indicated that participants' opinions of the CommandMap improved after they were given the opportunity to use it. In particular, participants underestimated their ability to remember control locations and interact quickly with the CommandMap (Table 2). This result was reinforced by subjects' comments, for example: "*at first glance I thought that the interface was too busy to find items. But later I found that I could remember icons exactly without the need*

*to look for them. I could use the interface when I pushed <Ctrl> so the interface was not on the screen all the time and it was easy to push or release the <Ctrl> button."* There was also a marginal effect ( $p = 0.05$ ) in the change of response to the question *I would be discouraged from using software with this kind of interface in the future*. Initially, the responses expressed disagreement (mean 2.0, s.d. 0.9); and after using the system their level of disagreement was stronger (mean 1.6, s.d. 0.8). There was no change in perceived ease of finding controls, or the degree to which the interface was seen as visually overwhelming.

#### STUDY TWO

Study One showed that user performance and subjective preferences for CommandMaps exceeds that of the standard interfaces when executing repetitive tasks that are interwoven with realistic activities on a document. This extends prior results showing that command selection time (without any associated task) is reduced with CommandMaps. In Study Two, we move our investigation further toward real-world use, adding the following elements of realism.

*Different application contexts.* CommandMaps are designed to be a general technique, so our second study examines both word-processing (Microsoft Word) and painting (Pinta) applications.

*Realistic learning curve.* Previous CommandMap studies have all used highly repetitive selections of a small number of commands within short duration experiments. Study Two, in contrast, examines a week of daily skill development in a more realistic task. Further, to gain better insight into novice interaction (and unlike prior studies), no explicit initial training was given.

*Less-repetitive tasks.* Unlike previous studies, most of the commands used in Study Two were executed only once per task, and there was a wider range of commands. Furthermore, the tasks were designed to intersperse typing and drawing activities between command selections.

*Open-ended tasks.* Study One gave participants a specific set of instructions, relieving the participant from the need to expend cognitive resources on issues such as task strategy; in Study Two, in contrast, participants were simply given a target document or image and asked to reproduce it.

#### Participants and Apparatus

Twelve paid participants (10 male, 2 female; aged 21-34 years) were recruited from a local university. None had



What is forestry? Forestry is the knowledge of the forest. In particular, it is the art of handling the forest so that it will render whatever service is required of it without being impoverished or overgrown. For example, a forest may be handled so as to produce saw logs, kiln-dried poles, barrel hoops, firewood, tan bark, or turpentine. The main purpose of its treatment may be to prevent the nesting of pests, to regulate the flow of streams, to support cattle or sheep, or it may be handled so as to supply a wide range and combination of uses. Forestry is the art of producing from the forest whatever it can yield for the service of man.

Before we can understand forestry, certain facts about the forest itself must be kept in mind. A forest is not a mere collection of individual trees, just as a city is not a mere collection of unrelated men and women, or a nation like ours merely a certain number of independent social groups. A forest, like a city, is a complex community with a definite reason. It has a soil and an atmosphere of its own, chemically and physically different from any other, with plants and shrubs as well as trees which are peculiar to it. It has a constant population of insects and higher animals entirely distinct from those outside. Most important of all, from the forester's point of view, the members of the forest live in an exact and intricate system of competition and mutual assistance, of help or harm, which extends to all the inhabitants of this complicated city of trees.

The trees in a forest are all helped by mutually protecting each other against high winds, and by producing a richer and moister soil than would be possible if the trees stood singly and apart. They compete among themselves by their need for support in the soil, and for light and space by the growth of their crowns in height and breadth. Perhaps the strongest weapon which trees have against each other is growth in height. In certain species, instead of shade, the tree which is overtopped has lost the race for good. The number of young trees which decay each other in this fierce struggle for existence is prodigious, so that often a few score per acre are all that survive to

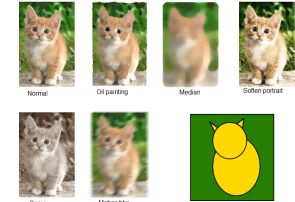
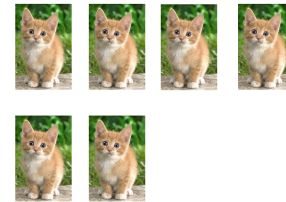
#### Some text about Forestry

What is forestry? Forestry is the knowledge of the forest. In particular, it is the art of handling the forest so that it will render whatever service is required of it without being impoverished or overgrown. For example, a forest may be handled so as to produce saw logs, kiln-dried poles, barrel hoops, firewood, tan bark, or turpentine. The main purpose of its treatment may be to prevent the nesting of pests, to regulate the flow of streams, to support cattle or sheep, or it may be handled so as to supply a wide range and combination of uses. Forestry is the art of producing from the forest whatever it can yield for the service of man.

This is a test box.

Before we can understand forestry, certain facts about the forest itself must be kept in mind. A forest is not a mere collection of individual trees, just as a city is not a mere collection of unrelated men and women, or a nation like ours merely a certain number of independent social groups. A forest, like a city, is a complex community with a definite reason. It has a soil and an atmosphere of its own, chemically and physically different from any other, with plants and shrubs as well as trees which are peculiar to it. It has a constant population of insects and higher animals entirely distinct from those outside. Most important of all, from the forester's point of view, the members of the forest live in an exact and intricate system of competition and mutual assistance, of help or harm, which extends to all the inhabitants of this complicated city of trees.

The trees in a forest are all helped by mutually protecting each other against high winds, and by producing a richer and moister soil than would be possible if the trees stood singly and apart. They compete among themselves by their need for support in the soil, and for light and space by the growth of their crowns in height and breadth. Perhaps the strongest weapon which trees have against each other is growth in height. In certain species, instead of shade, the tree which is overtopped has lost the race for good. The number of young trees which decay each other in this fierce struggle for existence is prodigious, so that often a few score per acre are all that survive to



a) Word Task 1 (before)

b) Word Task 1 (after)

c) Pinta Task 1 (before)

d) Pinta Task 1 (after)

Figure 3. Initial and target states for two of the tasks in Study Two.

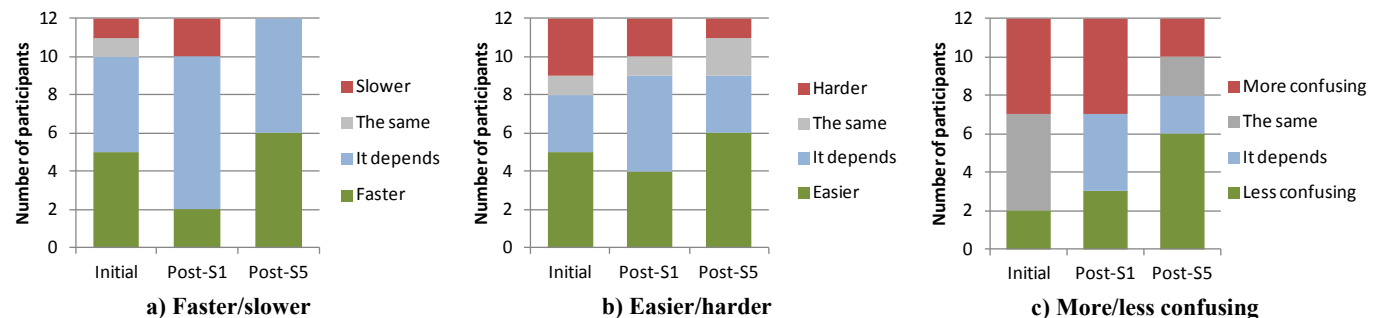


Figure 4. Participants' opinions before the experiment, after Session One, and after Session Five, on whether CommandMaps are faster or slower, easier or harder to use, and more or less confusing than the standard interfaces.

been involved in Study One, nor had any previously seen a CommandMap. Apparatus was the same as Study One.

### Procedure

Participants were assigned four tasks to complete on each of five successive days. Two tasks were completed using Word, and two using Pinta. One of the tasks with each application was performed using the CommandMap version of the software, and the other task was completed using the application's standard interface. To facilitate rigor in the analysis of skill development over the week, the experimental task with each interface was held constant for each of the five days. All participants completed the two tasks with each application in the same order, but the order of the applications and of their versions (standard and CommandMap) was counterbalanced between participants. Each of the four tasks took 5-10 minutes to complete.

For each Word task, participants were provided with an on-screen document containing text, as well as a printout of the intended formatting (see Figure 3). In the Pinta tasks, participants were given an initial image and had to apply filters and use drawing tools to replicate a target state shown on a printout. Participants were instructed to complete tasks in any way they wished, and to click 'Start' and 'Stop' buttons at the beginning and end of the task.

Participants completed a demographics questionnaire before the experiment, and filled out NASA-TLX [7] worksheets after each task. We gave additional questionnaires probing opinions of CommandMaps at the beginning and end of the first day (Session One) and at the end of Session Five. The participants were interviewed after completing all tasks on the fifth day. All sessions were videotaped.

### Results

Study Two was intended to answer research questions Q2 to Q5. We therefore present our results in terms of those questions, drawing on quantitative performance and preference measures as well as qualitative data from the video transcripts, questionnaires and post-study interviews.

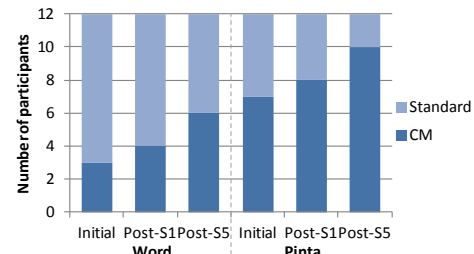


Figure 5. Participants' interface preferences for Word and Pinta throughout Study Two.

**Q2.How do users respond when first using CommandMaps?**  
Initial reactions to the Word and Pinta CommandMaps were mixed (see Figures 4 and 5, leftmost bars). One participant expected the CommandMap to slow his performance, but almost all others thought they would be faster at least some of the time. Measures of anticipated ease of use and confusion were less favorable, with three users anticipating the CommandMap to be harder than the normal interface, and five users rating it as initially more confusing. Eight participants commented that when they first saw the CommandMap, they found it overwhelming; but all of those participants said that their initial impression changed after only one or two five-minute sessions. For example, P3 said "when I first pulled it up, it was like 'whoa'... but then you quickly figure out the categories and then it's pretty quick." As seen in Figure 5, most participants initially

preferred the normal interface for Word, though six participants explicitly mentioned that the main factor in their decision was their existing familiarity with the Word interface. In comparison, with Pinta – an application that participants were less familiar with – the majority of participants (7/12) initially preferred the CommandMap.

Despite initial impressions, six participants commented in interviews that CommandMaps would be most advantageous for new users, since they provide an overview of an application's functionality. P1 thought that the main advantage of CommandMaps was “*getting to know the commands that are in a new application quickly*”, and P4 stated that “*one of the advantages is showing options that you maybe might not have known about*”. P1 indicated that he would recommend the interface to his mother, a novice Word user: “*I'd say 'just hold the Ctrl key and look, and you'll find it in there.' And that would help her, I think.*”

Two participants also commented that they were able to transfer their existing spatial knowledge of the Ribbon to the Word CommandMap: P8 stated that “*I could apply my existing knowledge to the layout of the CM, so I knew what section it was under and roughly how far along it was.*”

### Q3. Do performance and subjective impressions change...?

As expected, performance improved throughout the week in all four interfaces (Figure 6). We conducted  $2 \times 5$  ANOVAs on the Word and Pinta data for within-subjects factors *Interface* (standard, commandmap) and *Session* (1 – 5). As Figure 6 suggests, there was no effect of *Interface*, which is unsurprising given the relatively free-form method.

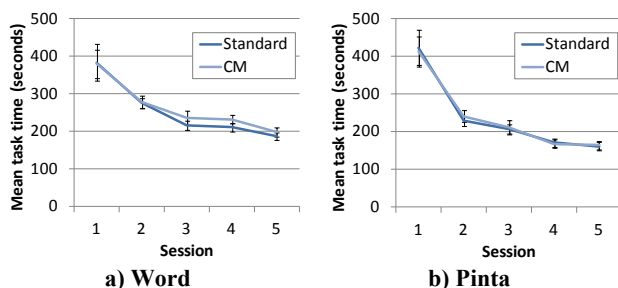


Figure 6. Mean task times for each session in Study Two.

Subjective impressions of the CommandMaps improved through the week (Figures 4 and 5). By the end of the week, 10/12 participants preferred CommandMaps for Pinta ( $\chi^2 = 4.1$ ,  $p = 0.04$ ), but preferences were evenly split for Word. Interview comments showed that several participants initially thought the CommandMap would be difficult to use, but changed their minds after trying it. Five participants mentioned that they liked the Pinta CommandMap because it removed submenus: “*once you get to know it, it's really fast. Like if you know the blur button is going to be here (points), it's basically Ctrl and then move your mouse and it's almost instantaneous, instead of going through the submenus.*” Two participants also liked the full-screen workspace with CommandMaps.

### Q4. Are CMs successful in multiple application contexts?

Figure 5 shows that CommandMap preferences were stronger in Pinta than Word, with familiarity with Word's Ribbon being mentioned by six participants in explaining their preferences. The two participants who maintained a preference for Pinta stated that its menu hierarchy made it easier to find commands compared to the CommandMap's parallelized layout. The perceived advantages of CommandMaps were the speed of accessing commands (five participants), not having to scroll through Ribbon tabs (one participant), and the ability to see an overview of all commands in the application (six participants).

### Q5. What usability issues do CommandMaps encounter?

Seven participants mentioned that they had problems using the Control key to activate the CommandMap. In particular, the CommandMap would briefly appear when issuing hotkey shortcuts: “*Once I finish a sentence, I automatically save. It's nothing I even think about, and [the CommandMap] flickers on screen.*” Three participants also noted that they would prefer the activation key to act as a toggle, rather than needing to hold it for continual display.

Nine participants mentioned that item layout was critical to both interfaces' success, with three stating that the Word Ribbon categories were poorly named. Four indicated that the Pinta CommandMap needed an improved layout, and P3 suggested placing category labels on the left-hand side of the CommandMap to improve category identification. Several participants disliked having Pinta's toolbox in the CommandMap: P5 said that “*when you're rapidly changing between different tools, I don't think CommandMaps are useful... I think you should be able to determine based on frequency which commands should be in a CommandMap and which should be permanently available*”.

Prior to the study we were concerned that the modal appearance of the CommandMap, hiding nearly the entire view of the work surface, might disorient users and interfere with their task flow. However, apart from flicker when using keyboard shortcuts, only two participants mentioned this as an issue. Five participants explicitly said it was not a problem, with several mentioning that it was better than the standard interface; P8 said “*I think it was more effective,*” P4 said “*with Word, you have to go to a different tab, and go back and do something, which can be a bit disorienting... whereas with this it's always the same thing,*” and P12 said “*You were kind of still looking in the same place as well, because it's right in the middle of the screen. I found I didn't lose track of where I was, whereas if I went back to the menu and had to go back down again I'd find that more distracting*”. The semi-opaque CommandMap display also received positive comments: P3 said “*I liked with the Pinta one, it was kind of transparent, so you could push Ctrl and still see what's underneath.*”

## DISCUSSION

To recap, Study One demonstrated that CommandMaps have performance advantages over standard interfaces in a



task that involved interweaving command invocation with actions on the work surface. Study Two analyzed the evolution of CommandMap perceptions over a week, with positive results: 7/12 participants immediately preferred the Pinta CommandMap to traditional interface controls, and 10/12 preferred it after five days of use; and half preferred the Word CommandMap to the highly refined and familiar Ribbon interface after just a few sessions of use.

### **Overcoming initial reactions**

We believe that commercial developers should consider deploying CommandMaps. However, the key cause for concern in doing so is that users may have a negative initial reaction to the substantially changed command layout. Participant comments lend some support to this: eight of the 12 Study Two participants made some sort of reference to an initial impression of visual overload (though this dissipated after a few minutes of use). However, this concern is one that is often encountered when a vendor wishes to improve the user's experience through interface revisions [10]. Primary tools for overcoming such concerns are advertising and consumer education. CommandMaps can also be designed to maximize knowledge transfer from existing interfaces. For example, our Word CommandMap maintained item layout within each category group, and the display of the Home tab was identical across the CommandMap and traditional interface.

### **Design issues and improvements**

#### *CommandMap invocation*

In our systems, the CommandMap was displayed by pressing the Control key, which caused minor usability problems for several of our participants, particularly when issuing keyboard shortcuts. Re-purposing a less-frequently used key such as Alt as the activation key would reduce the incidence of overloaded interface actions. An attractive supplemental method for activating the CommandMap is to use a dedicated button on a mouse, which would allow for unimanual interaction. Many mice support configurable buttons, and repurposing one of these for CommandMap activation would be a natural choice, particularly if CommandMaps were widely deployed.

#### *CommandMap hotkeys*

Pursuing the opposite idea to including CommandMap invocation on the mouse, there are interesting possibilities in moving CommandMap item selection onto the keyboard. Recent studies demonstrated the effectiveness of ExposeHK [16], which overlays visual controls with their hotkeys when a modifier key is pressed. CommandMaps would allow all hotkeys for the interface to be browsed at once, possibly with the CommandMap invoked by pressing Ctrl, and the hotkey overlay by additionally pressing Alt. Command selection could then be completed either by clicking as normal with the mouse, or by pressing the associated hotkey. This should also improve the learnability of hotkeys over the Ribbon, by allowing hotkey browsing without to the need to first move the cursor over the target.

### **In what situations are CommandMaps most useful?**

CommandMaps provide two primary advantages over traditional methods for organizing commands in graphical user interfaces, such as menus and ribbons. First, they help novice or intermediate users gain an overview of an application's functionality. Second, they increase the efficiency of expert users who have learned item locations. We therefore advocate for the use of CommandMaps in any situation where the command set is small enough to fit everything on-screen at once.

In applications with extremely large command sets (such as Autodesk Maya or 3ds Max), providing a CommandMap containing all of the functionality of an application may be infeasible. In these circumstances, there are other ways in which a CommandMap could be used to improve performance. First, it could provide a broad top-level hierarchy to the command set, similar to Maya's Hotbox [12]. Second, it could be provided as a shortcut interface, allowing the user to customize the CommandMap with the tools they most frequently use. Alternatively, it could contain a subset of the controls most frequently needed by novice users, providing access to a basic overview of the application's functionality.

### **Lessons for menu design**

Several studies of menu selection time with hierarchies of different breadth and depth have shown that selection times can follow a U-shape with depth, suggesting that menus should be broad, but not too broad [18]. However, the success of CommandMaps suggests that selection times can be reduced by making the hierarchy as broad as possible, reducing hierarchy depth to a single level. The combined results of this paper and [23] support Cockburn and Gutwin's [2] assertion that when users know item locations or can predict them based on some structure (e.g., alphabetical ordering or semantic categories), performance is optimized by flat command structures; studies such as [18] showed different results because they used random command arrangements and did not allow users enough time to develop item location memory. We refer the reader to [2] for a more complete analysis of menu layout studies.

More generally, these studies support the idea that spatial memory should be prioritized over visual search when optimizing control layouts. Current cognitive theory [4] shows that visual search is the primary location mechanism for novice users, and that retrieval from spatial memory is used when available. However, spatial memory develops rapidly [4, 11, 23] and as a consequence, users spend more of their lifetime with a software product retrieving locations from memory rather than searching for them. The interested reader should refer to [22] for more information on the theoretical aspects of spatial memory in HCI.

Another surprising outcome of Study Two was that users did not consider the modal nature of the CommandMap to be an impediment to use. This implies that with careful design (i.e. using translucency and a quasi-modal activation

key), future interfaces can successfully use a modal separation between content and interface controls to allow more screen real estate to be devoted to both.

### Supporting future implementations

A field study of CommandMaps is the next logical step in this project. However, in order to study their use ‘in the wild’, the technique would ideally be integrated with widely used software. Our experiences with developing a CommandMap version of Word suggest that achieving this integration can be difficult due to the lack of extensibility and support for modification in many commercial applications. For example, the Word Object Model API provided access to limited functionality, prohibiting our complete replication of the original Ribbon interface. Pinta, in contrast, was much easier to extend, but its small user base would limit the value of its use in a field study. In order to provide support in future for CommandMaps and similar HCI research, it would be useful if UI toolkits allowed manipulation of control and panel locations, as well as providing better support for programmatically accessing interface functionality. Metisse [1] is one example of a partial workaround created by the research community, though it only works on the X window system.

### CONCLUSION

The CommandMap is a user interface command selection technique that makes all, or nearly all, commands available to the user in a single display that is overlaid on the workspace when a modifier key is pressed. A previous study demonstrated the potential of CommandMaps by showing that the time to acquire a CommandMap item in response to a cued stimulus is faster than with menus or ribbons. However, critical unanswered questions remained regarding their practicality. To examine issues affecting the use of CommandMaps in realistic tasks, we implemented two real-world versions of CommandMaps for Microsoft Word and Pinta. Our first study showed advantages of CommandMaps over the Ribbon in tasks that involved interleaved access to the workspace and the CommandMap. Our second study showed how users’ perceptions of CommandMaps rapidly evolve over the first week of use, as well as highlighting their advantages and opportunities for improvement. In future work, we will study how best to arrange commands in full-screen interfaces, and develop further CommandMap systems for use in field studies.

### REFERENCES

1. Chapuis, O. and Roussel, N. Metisse is not a 3D desktop! in *UIST '05*, ACM, NY, USA, 2005, 13-22.
2. Cockburn, A. and Gutwin, C. A Predictive Model of Human Performance with Scrolling and Hierarchical Lists. *Hum.-Comp. Interaction* 24, 3 (2009), 273-314.
3. Cockburn, A., Gutwin, C. and Alexander, J. Faster Document Navigation with Space-Filling Thumbnails. in *Proc. CHI '06*, ACM Press, (2006), 1-10.
4. Cockburn, A., Gutwin, C. and Greenberg, S. A Predictive Model of Menu Performance. in *Proc. CHI '07*, ACM Press, (2007), 627-636.
5. Duh, H.B.-L., Tan, G.C.B. and Chen, V.H.-h. Usability evaluation for mobile device: a comparison of laboratory and field tests. in *MobileHCI '06*, 181-186.
6. Gutwin, C. and Cockburn, A. Improving List Revistation with ListMaps. in *Proc. AVI '06*, 396-403.
7. Hart, S. and Staveland, L. Development of NASA-TLX. in *Human Mental Workload*, 1988, 139-183.
8. Hertzum, M. User testing in industry: A case study of laboratory, workshop, and field tests. in *User interfaces for all. Proc. 5. ERCIM workshop*, (1999), 59-72.
9. Hochheiser, H. and Shneiderman, B. Performance Benefits of Simultaneous Over Sequential Menus. *IJHCI* 12, 2 (2000), 173-192.
10. Kabay, M.E. A discussion of the Microsoft office fluent user interface. *Ubiquity* 2008, March (2008), 1-4.
11. Kaptelinin, V. Item Recognition in Menu Selection: The Effect of Practice. in *InterCHI '93*, 183-184.
12. Kurtenbach, G., Fitzmaurice, G.W., Owen, R.N. and Baudel, T. The Hotbox: efficient access to a large number of menu-items. in *CHI '99*, ACM, 231-237.
13. Lafreniere, B., Bunt, A., Whissell, J.S., Clarke, C.L.A. and Terry, M. Characterizing large-scale use of a direct manipulation application in the wild. in *GI 2010*, 11-18.
14. Mackay, W.E. and Fayard, A.-L. HCI, natural science and design: a framework for triangulation across disciplines. in *DIS '97*, ACM, Amsterdam, 223-234.
15. MacKenzie, I.S. *HCI: An Empirical Research Perspective*. Elsevier Science, 2012.
16. Malacria, S., Bailly, G., Harrison, J., Cockburn, A. and Gutwin, C. Promoting hotkey use through rehearsal with ExposeHK. in *CHI '13*, ACM, Paris, 573-582.
17. McGrath, J. Methodology Matters: Doing Research in the Behavioral and Social Sciences. in *Readings in Hum.-Comp. Interaction: Towards the Year 2000*, 1995.
18. Miller, D. The depth/breadth tradeoff in hierarchical computer menus. in *Proc. HFES*, (1981), 296-300.
19. Newell, A. and Rosenbloom, P.S. Mechanisms of Skill Acquisition and the Law of Practice. in Anderson, J. ed. *Cog. Skills & Acquisition*, Erlbaum, 1981, 1-55.
20. Nielsen, C.M., Overgaard, M., Pedersen, M.B., Stage, J. and Stenild, S. It's worth the hassle!: the added value of evaluating the usability of mobile systems in the field. in *NordiCHI '06*, ACM, Oslo, Norway, 2006, 272-280.
21. Robertson, G., Czerwinski, M., Larson, K., Robbins, D., Thiel, D. and van Dantzich, M. Data Mountain. In *UIST '98*, (1998), 153-162.
22. Scarr, J., Cockburn, A. and Gutwin, C. Supporting and exploiting spatial memory in user interfaces. *Foundations and Trends® in HCI* 6, 1 (2013), 1-84.
23. Scarr, J., Cockburn, A., Gutwin, C. and Bunt, A. Improving command selection with CommandMaps. in *CHI '12*, ACM, Austin, Texas, USA, 2012, 257-266.
24. Shneiderman, B. Direct Manipulation: A Step Beyond Programming Languages (excerpt). in *Readings in HCI: A Multidisciplinary Approach*, 1987, 461-467.