

Supporting Feature Awareness and Improving Performance with Personalized Graphical User Interfaces

by

Leah Findlater

B.Sc., The University of Regina, 2001

M.Sc., The University of British Columbia, 2004

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE STUDIES
(Computer Science)

The University Of British Columbia
(Vancouver)

July 2009

© Leah Findlater, 2009

Abstract

Personalized graphical user interfaces have the potential to reduce visual complexity and improve efficiency by modifying the interface to better suit an individual user's needs. Working in a personalized interface can make users faster, more accurate and more satisfied; in practice, however, personalization also comes with costs, such as a reliance on user effort to control the personalization, or the introduction of spatial instability when interface items are reorganized automatically. We conducted a series of studies to examine both the costs and benefits of personalization, and to identify techniques and contexts that would be the most likely to provide an overall benefit.

We first interviewed long-term users of a software application that provides adaptable (user-controlled) personalization. A design trade-off that emerged is that while personalization can increase the accessibility of features useful to a user's current task, it may in turn negatively impact the user's awareness of the full set of available features. To assess this potential trade-off, we introduced awareness as an evaluation metric to be used alongside more standard performance measures and we ran a series of three studies to understand how awareness relates to core task performance. These studies used two different measures to assess awareness, showing that personalization can impact both the recognition rate of unused features in the interface and user performance on new tasks requiring those features. We investigated both adaptive (system-controlled) and adaptable personalization techniques to help us understand the generalizability of the awareness concept.

In addition to introducing and incorporating awareness into our evaluations, we studied how specific contextual and design characteristics impact the user's experience with adaptive interfaces. In one study, we evaluated the impact of screen size on performance and user satisfaction with adaptive split menus. Results showed that the performance and satisfaction benefits of spatially reorganizing items in the interface are more likely to outweigh the costs when screen size is small. We also introduced a new adaptive personalization technique that maintains spatial stability, called ephemeral adaptation, and evaluated it through two studies. Ephemeral adaptation improves performance over both another closely related adaptive technique and a traditional interface.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Tables	viii
List of Figures	ix
Acknowledgements	x
Dedication	xii
Statement of Co-Authorship	xiii
1 Introduction	1
1.1 Motivation	2
1.2 Thesis Goals	4
1.3 Thesis Approach and Overview	4
1.3.1 Subjective Response to Coarse-Grained Personalization	5
1.3.2 Awareness and Core Task Performance	5
1.3.3 Core Task Performance with Adaptive Personalization	6
1.3.4 Summary	7
1.4 Summary of Thesis Contributions	8
1.5 Thesis Outline	8
2 Related Work	10
2.1 Categories of Personalization	10
2.2 Challenges and Motivations for GUI Personalization	11
2.3 GUI Personalization Approaches and Evaluations	13
2.3.1 Evaluation Considerations	13
2.3.2 Adaptive Approaches	14
2.3.3 Adaptable Approaches	16

2.3.4	Mixed-Initiative Approaches	17
2.3.5	Modelling Performance	17
2.4	Designing for Learnability	18
2.5	Summary	20
3	Interview Study: Evaluating a Role-Based Personalization Approach	21
3.1	Introduction and Motivation	21
3.2	IBM Rational Application Developer	22
3.3	Interview Methodology	24
3.4	Findings	24
3.4.1	Overall Personalization Practice	24
3.4.2	Challenges in Coarse-Grained Personalization	25
3.4.3	Summary of Design Implications	28
3.4.4	Limitations of the Study	28
3.5	Conclusion	29
4	Layered Interface Study: Measuring Feature Awareness and Core Task Performance 30	30
4.1	Introduction and Motivation	30
4.2	Introducing and Defining Awareness	32
4.3	Experimental Methodology	33
4.3.1	Interviews to Define Command Sets	33
4.3.2	Conditions	33
4.3.3	Task	35
4.3.4	Design, Participants and Apparatus	36
4.3.5	Procedure	37
4.3.6	Measures	38
4.3.7	Hypotheses	39
4.4	Results	40
4.4.1	Core Task Performance	40
4.4.2	Awareness	41
4.4.3	Timeouts, Exploration, and Errors	42
4.4.4	Subjective Responses: Questionnaires and Interviews	42
4.4.5	Summary	43
4.5	Discussion and Future Work	43
4.6	Conclusion	44
5	Screen Size Study: Increasing the Benefit of Spatial Adaptation	46
5.1	Introduction and Motivation	46

5.2	Background	48
5.2.1	Adaptive Interfaces for Small Screens	48
5.2.2	Accuracy of Adaptive Personalization	48
5.3	Experimental Methodology	49
5.3.1	Conditions	49
5.3.2	Task	52
5.3.3	Quantitative and Qualitative Measures	53
5.3.4	Design	54
5.3.5	Participants	54
5.3.6	Apparatus	54
5.3.7	Procedure	54
5.3.8	Hypotheses	55
5.4	Results	55
5.4.1	Core Task Performance	56
5.4.2	Awareness	58
5.4.3	Errors	59
5.4.4	Subjective Measures	59
5.4.5	Summary	60
5.5	Discussion	61
5.5.1	Limitations of the Experiment	63
5.6	Conclusion	64
6	New Task Study and the Design Space of Personalized GUIs	65
6.1	Introduction and Motivation	65
6.2	Experimental Methodology	66
6.2.1	Conditions	66
6.2.2	Task	68
6.2.3	Design, Participants and Apparatus	69
6.2.4	Quantitative and Qualitative Measures	69
6.2.5	Procedure	70
6.2.6	Hypotheses	70
6.3	Results	70
6.3.1	New Task Performance	70
6.3.2	Awareness Recognition Rate	71
6.3.3	Core Task Performance	72
6.3.4	Errors	72
6.3.5	Subjective Measures	73

6.3.6	Summary	73
6.4	Discussion of New Task Study	74
6.5	Personalization Factors Affecting Performance and Awareness	75
6.5.1	Control of Personalization	75
6.5.2	Granularity	77
6.5.3	Visibility of Change	77
6.5.4	Frequency of Change	78
6.6	Design Implications	78
6.7	Limitations	80
6.8	Conclusion	80
7	Ephemeral Adaptation: Using Gradual Onset to Improve Menu Selection Performance	82
7.1	Introduction and Motivation	82
7.2	Ephemeral Adaptation	84
7.2.1	Abrupt Onset and Potential Benefit for Adaptive GUIs	85
7.2.2	Pilot Testing of Early Designs	85
7.2.3	Final Technique	86
7.3	Ephemeral Study 1: Proof of Concept	86
7.3.1	Experimental Methodology	86
7.3.2	Results	90
7.3.3	Summary and Discussion	93
7.4	Ephemeral Study 2: Ephemeral Adaptation Versus Adaptive Highlighting	93
7.4.1	Experimental Methodology	94
7.4.2	Results	95
7.4.3	Summary	97
7.5	Discussion	98
7.6	Applications for Ephemeral Adaptation	99
7.7	Future Work	99
7.8	Conclusion	100
8	Conclusion	102
8.1	Thesis Contributions	102
8.1.1	Identification of Challenges in Using a Role-Based, Coarse-Grained Personalization Approach	102
8.1.2	Awareness and Core Task Performance	103
8.1.3	Cost/Benefit of Adaptive Technique Characteristics	104
8.2	Directions for Future Research	105

8.2.1	Further Work on Awareness	105
8.2.2	Further Work on GUI Personalization in General	107
8.3	Concluding Comments	108
Bibliography		109
Appendices		118
A	Interview Study Materials	119
B	Layered Interface Study Materials	124
C	Screen Size Study Materials	138
D	New Task Study Materials	145
E	Ephemeral Adaptation Study Materials	151
F	UBC Research Ethics Board Certificates	157

List of Tables

4.1	Breakdown of baseline menu and toolbar command set.	34
4.2	Detail on awareness recognition test scores.	41
5.1	Accuracy, predictability, and stability of adaptive conditions.	52
6.1	Detail on awareness recognition test scores.	72
6.2	Design space for personalized GUIs.	76

List of Figures

3.1	Screenshot of RAD's Welcome screen.	22
3.2	GUI personalization mechanisms in RAD.	23
4.1	Sample menus and toolbars from the experimental conditions.	34
4.2	Screenshot of experimental system in minimal interface layer.	36
4.3	Distribution of task commands in full interface layer.	37
4.4	Core task performance.	40
4.5	New task performance.	41
5.1	Screenshots of Small and Large screen experimental setups.	50
5.2	Base adaptive algorithm.	51
5.3	Core task performance.	56
5.4	Awareness recognition test scores.	58
5.5	Selection times of frequently versus infrequently selected items.	59
5.6	Subjective satisfaction.	60
6.1	Experimental interface with adaptive split menu and static menu.	67
6.2	New Task Study performance measures.	71
7.1	Ephemeral adaptation applied to menus.	83
7.2	Ephemeral Study 1 selection times per trial.	90
7.3	Ephemeral Study 1 predicted and non-predicted trials.	91
7.4	Ephemeral Study 1 satisfaction ratings.	92
7.5	Experimental interface showing colour highlighted menu.	94
7.6	Ephemeral Study 2 selection times per trial.	96
7.7	Ephemeral adaptation applied to a news website.	99

Acknowledgements

I would first like to thank Dr. Joanna McGrenere, my supervisor. She has guided me for many years, from being a master's student fresh out of undergrad to finishing my PhD. Joanna's enthusiasm and insight have been invaluable in shaping this research. I appreciate all the support and patience she has provided throughout this process (especially when I was preparing my first few conference talks!). I will be taking many lessons with me about both research and mentoring.

I would also like to thank my supervisory committee members, Dr. Peter Graf and Dr. Gail Murphy, who brought a broader perspective to my research. They provided advice at many critical junctures, especially on my thesis proposal. My examining committee members, Dr. Andy Cockburn, Dr. Ron Rensink, and Dr. Rick Kopak, all provided thoughtful comments that improved the quality of the final dissertation.

Many others have also contributed directly to this dissertation. Several colleagues at the IBM Toronto Lab were influential in early stages of this research, particularly David Modjeska, who always provided useful feedback, and Jen Hawkins, my mentor at IBM Centers for Advanced Studies. It was a pleasure working with Rebecca Hunt Newbury and Jessica Dawson, undergraduate research assistants who helped in running several of the lab studies. I would also like to thank Dr. Ron Rensink for sharing his expertise in visual cognition and seeding the idea for using gradual onset in the ephemeral adaptation technique.

Less tangible, though no less important has been the support of my colleagues in the Imager Laboratory. Dr. Kellogg Booth has provided me with much thoughtful advice about research and career over the years. Andrea Bunt, Rock Leung, Joel Lanir, Tony Tang and Garth Shoemaker have all spent many hours reading over drafts of papers, listening to practice talks, and generally helping me stay sane. The last few years would definitely not have been the same without Karyn Moffatt, who shared an office with me from the beginning and was always ready to talk about research ideas, read over paper drafts, or take a much needed break (thank you for all the caramel macchiatos!).

Graduate school would have been a much less enjoyable experience without my family and friends. My parents and siblings have always encouraged me and provided me with a balanced perspective during both the easy and more difficult times of the PhD. My friends have helped me in so many ways, getting me away from the computer, listening to me talk about research dilemmas,

and coming up with anti-procrastination plans. Thank you also to Jon, who constructed this sentence for me.

The research in this dissertation was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) and IBM Centers for Advanced Studies.

Dedication

To my parents.

Statement of Co-Authorship

All work in this dissertation was conducted under the supervision of Dr. Joanna McGrenere. Dr. David Modjeska also played a supervisory role for the Interview Study in Chapter 3.

I am the primary contributor to all aspects of this research, with the exception of the Ephemeral Studies in Chapter 7. For the Ephemeral Studies, Karyn Moffatt and I were equal contributors to the conception, research and data analysis, and Jessica Dawson ran participants and contributed feedback on the study procedure and manuscript drafts.

Large parts of Chapters 3 to 7 are updated versions of published papers or submitted manuscripts that were written primarily by me with feedback from all co-authors:

- An earlier version of parts of Chapter 3 was published as: Findlater, L., McGrenere, J., Modjeska, D., 2008. Evaluation of a role-based approach for customizing a complex development environment. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1267-1270.
- A earlier version of parts of Chapter 4 was published as: Findlater, L., McGrenere, J., 2007. Evaluating reduced-functionality interfaces according to feature findability and awareness. *Proceedings of IFIP Interact*, 592-605. Republished with kind permission of Springer Science+Business Media.
- An earlier version of parts of Chapter 5 was published as: Findlater, L., McGrenere, J., 2008. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1247-1256.
- Chapter 6 combined with parts of Chapter 4 is in submission to the International Journal of Human Computer Studies.
- An earlier version of parts of Chapter 7 was published as: Findlater, L., Moffatt, K., McGrenere, J., Dawson, J., 2009. Ephemeral adaptation: The use of gradual onset to improve menu selection performance. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1655-1664. [Best paper award; top 1% of submissions]

Chapter 1

Introduction

Feature-rich graphical user interfaces (GUIs) are found in applications from word processors to integrated development environments. These interfaces support a wide range of tasks and provide many necessary features, but they provide many more features than are needed by any typical individual user. The excess features can be problematic for novice users, who may be overwhelmed by the sheer number of options available to them in a feature-rich interface. The impact of this complexity even extends beyond novice users, since experts tend to use only a small subset of available features [79, 87].

To reduce complexity and improve interaction efficiency, GUI personalization approaches can modify the user interface to better suit an individual's pattern of use. With some of these approaches, the system controls the personalization. Adaptive split menus, for example, automatically copy the menu items most likely to be needed by a given user to the top of the menu to make them more easily accessible [104]. In contrast, personalization can also be fully under user control. Layered interfaces, for example, allow the user to switch between several interfaces to the application, choosing the one that best suits his or her needs at a given point in time [106]; novice users may begin working in an interface layer that contains only a small, core set of features before transitioning to a more complex layer. Some techniques, such as that used in the Microsoft Windows XP Start Menu, combine both user and system control over the personalization: with the Start Menu, the most frequently and recently used items are automatically copied from cascading submenus into the top level of the menu, but users can also specify the number of items that appear in the top level and whether some items should remain there permanently.

Throughout this dissertation, we use the term personalization to refer to GUI personalization approaches that are *adaptive* (system-controlled), *adaptable* (user-controlled), or *mixed-initiative* (a combination of the previous two). We also focus on lightweight GUI personalization, particularly of menus and toolbars, rather than on personalization of content (e.g., recommender systems [1] and personalized search [117]) or on end-user programming [89], which allows the user to personalize

the interface but often requires deeper technical expertise and effort.

1.1 Motivation

Working in a personalized interface has its advantages. Personalized interfaces can make novice users faster [23, 56], more accurate and more satisfied [23], and can even be preferred by a large portion of more experienced users [85]. Despite these benefits, however, researchers have also identified drawbacks of both adaptable and adaptive mechanisms. Adaptive mechanisms offload the burden of personalization to the system but introduce other costs associated with unpredictability, instability, and the need for trust in the system's ability to provide useful adaptations [62].

In contrast, one issue with adaptable mechanisms is that the inclusion of a personalization mechanism itself adds complexity, even while the goal is to reduce complexity [70]. Adaptable mechanisms also require effort on the part of the user, a barrier which users often do not overcome [80, 81]. To address the latter problem, a mixed-initiative approach that provides adaptive suggestions to help the user make efficient personalization choices has been shown to have potential [20]. Another tactic that should reduce the burden of effort required by the user is to use *coarse-grained* personalization: features are grouped based on some notion of similarity, and the user can enable or disable large groups of features at once, rather than having to enable or disable individual features separately. Layered interfaces and role-based personalization (tailoring the interface to a user's work role) are examples of this type of approach. Unfortunately, evaluation of coarse-grained approaches has been limited to relatively simple applications or customization models [29, 30, 39, 98, 106], so we do not fully understand their effectiveness.

Another issue with personalized interface research is that evaluations have focused on the benefits of personalization, while drawbacks have largely been ignored. One potential cost of personalization that has not yet been explored in the research literature is the effect that it can have on learnability. An underlying theme of previous work is that personalization itself should be inherently beneficial if we can design the mechanisms appropriately to mitigate the disadvantages described above. While we do not dispute that interface personalization can be beneficial, we also argue that working in a personalized interface can impact users in ways that are not necessarily captured by the traditional evaluation measures of user satisfaction and performance. When an interface is personalized to make it easier to access the features most useful to a user's core tasks, it may negatively impact the user's awareness of the full set of available features, features that might be potentially useful in the future. A survey of 53 users of Microsoft Word 97 reflected this tension between core task performance and the ability to learn about new features: while many users requested that their unused features be tucked away, many also indicated that it was important to be able to continually discover new features [87].

Considering personalization in terms of a cost/benefit trade-off will also help us to understand

why GUI personalization succeeds in some contexts but not in others. Personalizing an interface can positively impact performance, for example, by reducing navigation time to reach items and by reducing visual search time. However, researchers have only recently begun to explore what characteristics of the individual techniques, the application context, and the users themselves will lead to an overall performance benefit. Results have shown, for example, that adaptive approaches that maintain spatial stability by replicating menu and toolbar items are preferred to approaches that move items [48], and that the predictability of adaptive personalization can also impact user satisfaction [51]. This is particularly important for adaptive GUI personalization since previous research has yielded conflicting performance results [38, 50, 52, 51, 56, 88, 104].

Personalization techniques that spatially reorganize features in the interface should reduce navigation time, and, to a lesser degree, visual search time. However, spatial personalization also introduces instability into the interface layout, which can reduce potential gains. For adaptable menus, spatial reorganization has been shown to be as fast as an optimal static counterpart [38]. In contrast, spatially adaptive personalization approaches are not often faster than their static counterparts and may sometimes be slower [31, 38, 88]; successes have tended to occur when the personalization greatly reduces the number of steps to reach desired functionality [50, 56, 125]. One potentially fruitful application area that has not yet been explored is the use of adaptive GUI personalization for small screen devices. The reduced screen size means that, even with high resolution screens, designers must choose only the most important features to display, requiring navigation to reach off-screen items. Despite the potential theoretical benefits for smaller screens, research to date has focused largely on adaptive web content (e.g., [67, 111]) and has not evaluated adaptive GUI control structures.

Another approach to offering some of the benefits of personalization without the drawbacks of spatial reorganization is to focus only on reducing the user's visual search time to find items. Several researchers have proposed techniques to highlight predicted items with a different background colour [48, 50, 124] but no performance results comparing colour highlighting to a static control have been reported. Another possibility that has been underexplored is to use a temporal dimension to draw the user's attention to only a small subset of items in the interface by having those items appear abruptly, before fading in the remaining items gradually (e.g., in a pull-down menu). There is even some evidence in the human perception literature that abrupt onset of stimuli, where an item appears suddenly on the screen, may be a stronger cue for grabbing the user's attention than colour [120]. However, abrupt onset has not been previously explored in the context of user interface personalization.

1.2 Thesis Goals

At a high level, the goal of this thesis has been to improve the design of personalized graphical user interfaces. We used a three-fold approach: (1) studying user perceptions of personalization; (2) developing an evaluation methodology to more comprehensively capture both the costs and benefits of personalization than is possible with existing evaluation techniques; (3) identifying application contexts and design choices that best take advantage of these benefits.

More specifically, our goal was to answer the following questions:

1. From the user's point of view, what are positive and negative aspects of working in an interface that provides adaptable coarse-grained personalization?
2. Objectively, is there a trade-off between core (routine) task performance and the user's overall awareness of features in a personalized interface?
 - (a) How can we operationalize these concepts so a potential trade-off can be measured?
 - (b) If such a trade-off does exist, does the reduced awareness result in a measurable impact on performance when users are asked to complete new tasks?
3. How can we increase the likelihood that the potential benefits of adaptive personalization for core task performance (e.g., reduced navigation and visual search time) will outweigh the associated costs (e.g., cognitive overhead of dealing with instability of interface layout)?
 - (a) Are personalization techniques that spatially reorganize items more effective when it is more difficult to view and navigate to items, such as when screen real estate is constrained?
 - (b) As an alternative to spatially reorganizing items, can a personalized interface outperform a static interface layout solely by reducing visual search time?

1.3 Thesis Approach and Overview

To satisfy the goals of this thesis, we took a multifaceted approach, including first an exploratory semi-structured interview study, followed by five controlled laboratory studies. The interview study allowed us to collect user feedback on the experience of working long-term with a commercial application that provides GUI personalization as a major component of its interaction. The remaining studies were motivated by a combination of the first interview study and a survey of related work, and build upon each other.

The work conducted for this thesis encompasses a variety of personalization techniques: Goal 1 focused specifically on adaptable, coarse-grained personalization; Goal 2 applied more broadly to

both adaptive and adaptable personalization; Goal 3 focused on adaptive personalization. Answering these questions should provide insight into why many personalization approaches have not been successful (e.g., the Microsoft Office 2003 adaptive menus). The results should provide researchers with fruitful directions for further work and allow developers to make an informed decision about what type of personalization, if any, would be useful in a given context.

1.3.1 Subjective Response to Coarse-Grained Personalization

To address the first thesis goal we conducted a preliminary set of interviews with 14 users of a complex integrated development environment that provides role-based personalization, where only those features associated with a user's work role are enabled in the interface (Chapter 3). The goal of the Interview Study was to explore the potential benefits and challenges of personalization approaches that reduce the number of features shown to the user through a coarse-grained personalization model. Of particular importance to the remainder of this dissertation research, we found that more than half the participants were concerned about hiding features, in part because it could impair their ability to learn about and use new features in the interface. This finding is consistent with previous work with word processor users [87].

1.3.2 Awareness and Core Task Performance

Based on the findings from the Interview Study, we explored the tension between personalizing to improve performance versus the user's ability to learn about new features, which was the second goal of this thesis. We defined *feature awareness* as a new evaluation measure: awareness is the degree to which users are conscious of the full set of available features in the application, including those features that have not yet been used. Awareness of features that have not yet been used is a measure of the secondary, incidental learning that may occur as the user performs a primary task. We proposed that measuring awareness in conjunction with core task performance will be particularly valuable for personalized interfaces. Together, these measures offer a broader understanding of the impact of working in a personalized interface. Although awareness, by definition, does not impact core task performance, it has the potential to impact performance on new tasks. We thus also distinguished core task performance from new task performance.

We conducted three controlled lab studies to demonstrate and characterize the trade-off between core task performance and awareness when working in a personalized interface. We operationalized awareness using two measures: (1) recognition rate of unused features in the interface, and (2) task performance when users are asked to complete new tasks that require those features. The first two studies (Layered Interface Study and Screen Size Study) each compared a different type of personalized interface to a control condition to demonstrate a trade-off between core task performance and awareness, when measured using our first method, the recognition rate of unused features.

The Layered Interface Study (Chapter 4) was a proof-of-concept study and compared minimal and marked layered interface conditions to a static control condition (the full interface to the application). The minimal layered interface condition improved core task performance over the control condition; however, subjects in the minimal layered condition were not as aware of advanced features after transitioning to the full interface as those who had worked in the full interface from the outset. The marked approach, in contrast, had no significant impact on either core task performance or awareness.

The Screen Size Study (Chapter 5) focused on adaptive split menus [104]. The study included 36 participants, who worked with a static control condition and two adaptive split menu conditions that predicted the user's needs with different degrees of accuracy (50% vs. 78%). Results showed that the higher accuracy split menu condition produced lower awareness recognition rates than either the static control or the lower accuracy condition, likely because it reduced the need for users to search through the entire menu to find a feature.

The third study incorporating awareness also focused on adaptive split menus, but extended results from the Layered Interface Study and the Screen Size Study to provide evidence that personalization impacts our second measure of awareness: performance on new tasks (Chapter 6). The conditions in the New Task Study were similar to the Screen Size Study. As expected, participants were fastest at completing new tasks in the control condition, while the higher accuracy adaptive condition provided the best core task performance. The low accuracy adaptive condition provided neither a core task nor a new task performance benefit over the control condition.

Drawing on a survey of related work and our own results, we also outlined a design space of GUI personalization (Chapter 6). We identified four design factors that are particularly important for the interplay between core task performance and awareness: control, granularity, visibility, and frequency of personalization. The design space allows us to identify fruitful areas for future work, and, combined with the study results, we build on it to present several design guidelines for applying personalization approaches.

1.3.3 Core Task Performance with Adaptive Personalization

In addition to introducing and incorporating awareness into our evaluations, we were interested in how specific contextual and design characteristics impact the user's experience with adaptive personalized interfaces. To address Goal 3a, the Screen Size Study mentioned above was also designed to evaluate the impact of screen size on performance, awareness, and user satisfaction with adaptive split menus. Then, to address Goal 3b, we introduced a new GUI personalization technique that maintains spatial stability, called *ephemeral adaptation*, and evaluated it through two additional studies. We revisit the Screen Size Study and describe the two additional studies next.

To explore how well previous findings on adaptive personalization generalize to small screen devices, the Screen Size Study compared adaptive interfaces for small and desktop-sized screens.

Results showed that adaptive split menus that predict the user's needs with high accuracy had an even larger positive impact on performance and satisfaction when screen real estate was constrained. As mentioned in Section 1.3.2, however, the drawback of the high accuracy menus is that they reduce the user's awareness of the full set of items. Overall, the findings stress the need to revisit previous adaptive interface research in the context of small screen devices.

Finally, to determine whether an adaptive personalization technique can provide performance improvements through aiding visual search alone, we introduced ephemeral adaptation. Ephemeral adaptive interfaces maintain spatial stability but employ gradual onset of GUI items to draw the user's attention to predicted ones: adaptively predicted items appear abruptly when the menu is opened, but non-predicted items fade in gradually. To demonstrate the benefit of ephemeral adaptation we conducted two studies with a total of 48 users. Ephemeral Study 1 refined the ephemeral adaptation technique. Ephemeral Study 2 then compared ephemeral adaptation to colour highlighting and showed that: (1) ephemeral adaptive menus are faster than static menus when the accuracy of the adaptive predictions is high, and are not significantly slower when it is low; and (2) ephemeral adaptive menus are also faster than adaptive highlighting. While we focused on user-adaptive GUIs, ephemeral adaptation should be applicable to a broad range of visually complex tasks.

1.3.4 Summary

This dissertation shows that personalization of graphical user interface control structures such as menus and toolbars can improve performance, especially when the personalization greatly reduces navigation time or improves visual search. However, we have also shown that personalization can negatively impact the user's awareness of unused features in the interface for both adaptive and adaptable techniques, which, in turn, results in poorer performance when users are asked to complete new tasks. These findings motivate the need to explicitly broaden the user's knowledge of unused features in a personalized interface, and highlight the potential of providing adaptive suggestions of new features to users (e.g., [79]).

Combined, the studies conducted for this dissertation cover a wide variety of GUI personalization mechanisms, all of which adapt the menu and toolbar content. This includes four previously developed approaches: layered interfaces [106], role-based personalization [95], adaptive split menus [38, 104], and adaptive colour highlighting [48, 50, 124]. We also evaluated two new approaches: a modified version of layered interfaces, called a marked layered interface, and ephemeral adaptation; results from the latter were more promising.

1.4 Summary of Thesis Contributions

The primary contributions of this thesis are:

1. Identification of challenges inherent in using a role-based (coarse-grained) approach to reduce interface complexity, through interviews with users of a commercial application that provides such personalization (Chapter 3).
2. Cost/benefit of awareness versus core task performance:
 - (a) Introduction of awareness as a new evaluation metric for personalized GUIs, to be used alongside core task performance and operationalized as: (1) the recognition rate of unused features in the interface; and (2) performance on tasks requiring previously unused features (Chapter 4).
 - (b) Empirical evidence that there is a measurable trade-off between core task performance and awareness, when measured using a recognition test of unused features. Results hold for at least two types of personalization: layered interfaces (Chapter 4) and adaptive split menus (Chapter 5).
 - (c) Empirical evidence that there is a measurable trade-off between core task performance and awareness, when measured as the time to complete new tasks (Chapter 6). This provides a low-cost indication of future performance.
 - (d) A design space for personalized GUIs, outlining how different points in the space affect the trade-off between core task performance and awareness, and a corresponding set of design guidelines (Chapter 6).
3. Cost/benefit of adaptive technique characteristics:
 - (a) Empirical evidence demonstrating the relative benefit of adaptive GUIs for small displays in comparison to large displays. This benefit is due to both a reduction in navigation required to reach items and a change in user behaviour (Chapter 5).
 - (b) A new adaptive interface technique, called ephemeral adaptation, that adapts the interface along a temporal dimension (Chapter 7).
 - (c) Empirical evidence to show that ephemeral adaptation improves performance and user satisfaction in comparison to a static alternative and a colour highlighting approach when adaptive accuracy is high (79%) (Chapter 7).

1.5 Thesis Outline

In Chapter 2 we begin by presenting related work. Chapter 3 discusses the results of the Interview Study, while Chapters 4-7 present controlled laboratory studies that focus on a variety of aspects

of personalization. Chapter 4 (Layered Interface Study) is the proof-of-concept study evaluating core task performance and awareness for layered interfaces. Chapter 5 (Screen Size Study) turns to another type of personalization, adaptive split menus, measuring both the impact of screen size and different levels of adaptive accuracy on performance, awareness, and user satisfaction. Chapter 6 (New Task Study) builds on the awareness findings from the previous two chapters to show that personalization can impact performance on new tasks. In Chapter 7, we introduce and evaluate ephemeral adaptation through two studies (Ephemeral Studies 1 and 2). Finally, Chapter 8 summarizes the thesis work and contributions, and outlines areas for future work. Following this chapter, several appendices provide the study materials for all six studies, along with UBC Behavioural Research Ethics Board approval certificates (Appendix F).

Previous versions of the majority of the work presented in this thesis have already been published: Chapter 3 at ACM CHI 2008 [41], large parts of Chapter 4 at IFIP Interact 2007 [39], Chapter 5 at ACM CHI 2008 [40], and Chapter 7 at ACM CHI 2009 [42]. Chapter 6 combined with parts of Chapter 4 is in submission to the International Journal of Human Computer Studies.

Chapter 2

Related Work

This chapter provides an overview of research on graphical user interface (GUI) personalization, outlining broad categories of personalization, commercial and research examples of personalized GUIs, and results from empirical evaluations. In addition, we briefly summarize research on learnability of user interfaces. Further detail on relevant literature also appears in the individual chapters to follow.

2.1 Categories of Personalization

Generally speaking, personalization to manage complexity can be grouped into two main categories: (1) personalization of content (e.g., intelligent tutoring systems [4, 18], reducing software code complexity [72], collaborative filtering [118], and recommender systems [26, 61]), and (2) personalization of GUI control structures (e.g., changing menu and toolbar options). While there can be some overlap between personalization of content and personalization of GUI control structures, such as personalized restructuring of hypermedia links [14], this dissertation focuses particularly on GUI personalization of menus and toolbars. We are also particularly interested in lightweight GUI customization mechanisms, in contrast to approaches such as tailorable systems (e.g., [113]) and end-user programming (e.g., [89]), which often require deeper advanced technical expertise or user effort. Non-programmers in particular face several challenges in doing end-user programming [73], although programming by demonstration, such as macros, may reduce the need for technical expertise (e.g., [34]).

Personalization of content has traditionally been more successful than personalization of GUI control structures. As examples, recommender systems (see [1] for a survey), personalized search (e.g., [117, 97]) and Intelligent Tutoring Systems (e.g. [4, 18]) have achieved relative success. One possible source of this difference is the size of the search space for content personalization: Amazon.com’s catalog, for example, contains millions of products, but personalized recommendations

provide the user with a small, manageable set of products to browse through; in contrast, a complex application has at most hundreds or a few thousand features, so the relative benefit of personalized interaction is different. Bunt [17] also identifies user expectations and the importance of consistency as factors that contribute to the relative success and acceptance of personalized content in comparison to personalized GUIs. Users expect content to be changing, whereas they expect the set of features provided by a GUI to remain constant. Consistent layout in a GUI can allow users to make use of spatial memory as they learn item locations, reducing visual search time and developing motor memory.

Another broad category of personalization not covered in this dissertation is aesthetic personalization to modify the appearance of user interfaces and devices. For example, applying a skin or changing the desktop wallpaper on a personal computer would fall into this category of personalization. Blom and Monk [11] have shown that aesthetic personalization can have cognitive, social and emotional effects on the user, including increased perceived ease of use and feeling of ownership.

We use the term personalization to cover adaptable (user-controlled), adaptive (system-controlled) and mixed-initiative (shared control) approaches. Even within this space there are overlapping terms used in the research literature. The term *customizable* is often used interchangeably with the term adaptable. As well, the term *intelligent user interface* is sometimes used interchangeably with adaptive interfaces, although it also applies more broadly to systems that incorporate artificial intelligence but are not adaptive to an individual user, for example, expert systems.

2.2 Challenges and Motivations for GUI Personalization

Adaptive and adaptable personalization approaches both have commonly cited advantages and disadvantages. Fischer [43] identifies the major strengths of adaptive interfaces to be that they require little or no effort on the part of the user and that the user is not required to have specialized knowledge about the system in order to adapt the interface. In contrast, adaptable approaches provide the benefits of direct manipulation interfaces in general, including predictability, controllability, and comprehensibility [105]. The user may also know his or her task better than adaptive reasoning can determine, so adaptable personalization will always be done with the goal of supporting the user's actual task [43].

Höök [62] summarizes several well-known issues with adaptive interfaces, similarly echoed by critics of adaptive interfaces (e.g., [107]). These issues include the lack of control the user has over the adaptive process, and the difficulty that users may have in predicting what the system's response will be to a user action. There is also the problem of transparency, that the user may not understand how the adaptivity works in the interface, and it is not clear how much of the underlying adaptive process should be exposed to the user. Providing rationale can lead to increased trust in the system, but not all users want the information [21] and it does not necessarily increase users'

understanding of the system [96]. Höök [62] also cites issues of privacy, because the system needs to collect information about the user to adapt accordingly, and issues of trust, because the user needs to trust that the adaptive behaviour accurately reflects his or her needs. Obtrusiveness, or the degree to which the interface is distracting or irritating to the user, has also been cited as an issue with adaptive interfaces [66].

Adaptable interfaces address some of these issues, but face other challenges. One major challenge is that adaptable interfaces require effort on the part of the user. The extent to which users adapt their interface is dependent on skill level and interest, with some users not making any adaptations at all [80, 82]. In a study of 51 users of a UNIX system, Mackay [81] found that customization was affected by external events (e.g., job changes), social pressure, software changes (e.g., upgrades), and internal factors (e.g., excess free time). Actual customization was minimal because of time, difficulty, and lack of interest. In a study of 101 users of a word processor, Page, Johnsgard, Albert and Allen [93] showed that almost all users (92%) did some form of customization, but their definition was broad and customizations included small changes such as showing or hiding the ruler bar. Another disadvantage of adaptable approaches is that the inclusion of the mechanism for the user to adapt the interface increases the complexity of the interface, and can be particularly problematic if that mechanism is poorly designed [70]. Page et al. [93] also showed that customization features that were simple to use tended to be used more often.

Oulasvirta and Blom [91] studied the motivations behind why users choose to personalize technology. Based on the theory of Self-Determination, they categorized motivations as being related to autonomy, competence and relatedness (social relation to other people), and distinguished between personalization of a user's own experience with a piece of technology and personalization that changes the technology's appearance to others. Based on motivations of autonomy and competence, for example, users will personalize their interface to make it more efficient and to increase enjoyment and engagement. The motivation of relatedness means that users will also personalize to express identity, emotion, and to convey competence with the technology. Although not the focus of the work, Oulasvirta and Blom also state that some of these motivations extend to adaptive systems. For example, providing adaptive recommendations as one of several means of accessing information increases autonomy; to use Gajos' [47] terminology, the goal is to make the adaptivity elective rather than mandatory.

Another issue that has received a small amount of attention in the literature is the impact that adaptive interfaces may have on the user's breadth of experience: when part or all of the user's task is offloaded to the system, the user may end up learning less [66]. The issue has arisen particularly in the context of recommender systems, where providing highly accurate recommendations (i.e., items the user is sure to like) may not be as satisfying to the user as lowering the accuracy but providing a wider breadth of recommendations [130]. Outside of recommender systems, we are not aware of research on how personalization impacts breadth of experience. By introducing awareness as a

novel evaluation metric, this dissertation addresses that gap in the context of GUI personalization, and is predicated on the belief that both adaptive and adaptable personalization mechanisms will impact awareness (Chapters 4, 5, 6).

Despite the challenges facing adaptive and adaptable personalization, the benefits have been shown to outweigh the costs in various contexts, including for both novice (e.g., [22, 23]) and expert [85] users, and for specific user groups, such as motor-impaired individuals [52]. Mixed-initiative approaches also have potential to bridge the advantages of both adaptive and adaptable approaches [20]. Another factor that needs to be considered is the user's perceived benefit of personalization, which can impact the effectiveness of both adaptive and adaptable approaches. For example, when the accuracy of an adaptive algorithm's predictions is higher, users are more likely to make use of correct predictions [124] (we also show this in the Screen Size Study, Chapter 5). Perceived benefit also impacts adaptable approaches: customizing a densely populated dialog box has resulted in a greater reduction in perceived workload than customizing a sparsely populated dialog box [101]. We highlight these findings and more in the following section.

2.3 GUI Personalization Approaches and Evaluations

We classify personalization approaches as being either spatial, graphical, or temporal: spatial approaches reorganize or resize items in the interface, graphical approaches visually distinguish items through distinct colours or other methods, and temporal approaches may use spatial and/or graphical techniques, but the adaptation only lasts briefly.

2.3.1 Evaluation Considerations

Evaluating personalized interfaces is more complex than evaluating traditional static interfaces, because there is increased variability as the interface changes over time. This has been noted particularly in the context of adaptive personalization [83], and several researchers advocate separating out evaluation of the underlying inference mechanism or user modelling component from evaluation of the interface adaptations that are made based on that information [15, 127]; the scope of this dissertation falls within the latter.

User characteristics are also important for assessing the overall effectiveness of personalization approaches to manage user interface complexity. For example, gender [7], personality traits related to control [53], and whether users are feature-keen or feature-shy with respect to having extra, unused items in the interface [85] may all affect the success of a system.

2.3.2 Adaptive Approaches

Adaptive GUI personalization has largely focused on spatial techniques, with a smaller amount of work on graphical and temporal techniques.

In early work, Greenberg and Witten [56] conducted a controlled lab study with 26 participants and showed that adaptively restructuring a hierarchical menu system based on frequency of use improved performance over the original static layout. In a follow-up study with 4 participants, however, Trevelyan and Browne [123] showed that as users gained experience the adaptive menu system did not offer a benefit over the static layout. An early negative result came from work by Mitchell and Shneiderman [88]. In an evaluation with 63 participants, they compared static to adaptive pull-down menus that reordered during usage based on frequency, finding the static menus were faster and preferred.

More recently, split interfaces have received a relatively large amount of research attention. Since being introduced in the form of split menus by Sears and Shneiderman [104], they have been extended to provide both adaptive and adaptable control. With a split menu, the items most likely to be needed by the user are either moved or copied to the top of the menu, above a separator line. Theoretically this should result in improved performance. An evaluation with 38 participants in the original split menu design, where items were moved to the adaptive top section of the menu, showed that split menus were at least as fast, or faster, than traditional static menus [104]. However, the layout of those menus was predetermined before use and subsequent evaluations of split interfaces have yielded conflicting results [31, 38, 50].

Findlater and McGrenere [38] conducted a study with 27 participants comparing adaptive split menus (where items were moved above the split) to static and adaptable split menus. The adaptive split menus were slower than both the static and adaptable ones, except, in the case of the latter, when adaptable appeared first in order of experimental presentation. In a study with 16 participants, Gajos et al. showed that users had a strong preference for an adaptive split interface that replicated items in the adaptive section, in comparison to a static counterpart [48], suggesting that replicating items may be better than moving them in split interfaces.

The personal menus in Microsoft Office versions 2000 to XP are an example of a commercial adaptive interface that is familiar to most people: when the user first opens the menu, only a subset of items are displayed, but after hovering or clicking the arrow at the bottom of the menu, the menu expands to reveal the full set of items. The menus were subsequently removed in Microsoft Office 2007, and empirical evaluation on a similar type of expanding adaptive menu showed that it is less efficient to use than a traditional static menu [44].

There are also examples of spatial adaptive techniques that resize items rather than moving them. For example, morphing menus [31] dynamically increase the size of the most frequently selected items and walking user interfaces [69] change the size of interface elements to address “situational impairments”, such as reduced-motor ability and attention while walking versus standing

still. Neither have so far been shown to provide performance improvements.

Gajos et al.'s SUPPLE system [49, 48] adjusts the layout, size and structure of interface elements to accommodate both device and user characteristics. It has recently been extended to adapt the interface to the needs of motor-impaired individuals, both through explicit preference elicitation from users and automatic detection of motor abilities [52]. An evaluation with 17 participants showed that adapting the interface based on automatic detection of motor abilities made motor-impaired users faster and more accurate than with a manufacturer's default interface layout, and was strongly preferred. This approach adapts the interface only once at the outset, but it does highlight a cost/benefit trade-off of personalization: while both motor-impaired and able-bodied users were faster with their personalized interfaces, the able-bodied users, who did not achieve as large a performance gain, preferred the aesthetics of the original interface layout.

A drawback of spatial adaptation techniques is that they introduce instability into the layout of the interface, potentially impacting performance. As mentioned above, Gajos et al. [48] showed that users prefer to have items copied rather than moved to the adaptive section of a split interface. This increases the overall stability of the interface layout and makes the adaptive predictions optional, decreasing the impact of incorrect predictions. In Chapter 5 we explore whether screen size impacts the cost/benefit trade-off of spatial adaptation.

In contrast to spatial adaptation techniques, graphical adaptation offers spatial stability, which could be particularly beneficial as users gain experience and spatial memory of item locations. There has, however, been far less research in this area. Colour highlighting of adaptively predicted items has been proposed by several researchers [48, 50, 124] to reduce visual search time but evaluations have either been inconclusive or have not included a static control condition. Another graphical technique is the bolding of adaptively predicted items, but an evaluation with 32 participants did not demonstrate a benefit over a static control condition [94]. Temporal adaptation techniques can also provide spatial stability, depending on the design. However, the only work we are aware of that has looked at temporal adaptation provided only minimal adaptation (adaptive predictions changed once during the course of use) and yielded negative results when evaluated with 16 participants [76]. We empirically evaluate adaptive colour highlighting and introduce a new and effective temporal adaptation technique in Chapter 7.

Spatial, graphical and temporal adaptive approaches all refer to how interface adaptations are presented to the user based on an underlying adaptive algorithm or user model. However, researchers have also begun to isolate how characteristics of the underlying algorithm impact the user's experience. Tsandilas and Schraefel showed, in a study with 12 participants, that the accuracy with which the adaptive algorithm can correctly predict the user's needs impacts performance: higher accuracy results in faster performance. Gajos et al. [50] subsequently found a similar result with adaptive split menus, in a study with 8 participants. From the user's point of view, the predictability of the adaptations can also impact satisfaction [51].

2.3.3 Adaptable Approaches

Many commercial and research examples of adaptable interfaces exist; for example, Microsoft Office XP allows users to specify the contents and layout of toolbars. To our knowledge, adaptable approaches have been limited to spatial adaptation.

Some personalization approaches provide preset functionality tailored to individual users' skill levels and in this way can be considered adaptable. An early example is the training wheels approach, first introduced by Carroll and Carrithers [23]. Novice users begin working in a training wheels interface, where advanced functionality is blocked with the goal of reducing error rates. Evaluation results with 12 participants showed that the training wheels approach makes novice users faster and more satisfied [23]. A follow-up study with 16 participants also showed that users who had learned on a training wheels interface were faster when asked to perform advanced tasks on the full interface than those who had used the full interface from the outset [25]. However, more recent research studying the use of training wheels for a modern word processor by 72 participants suggests that users may not see the value of such designs in a classroom setting [6]. Users were more satisfied with a full interface than they were with a training wheels interface; however, the presence of a tutor likely impacted the overall benefit of the training wheels interface.

Another approach that enables or disables large groups of menu and toolbar items at once is the layered interfaces approach, introduced by Shneiderman [106]. Layered interfaces provide several different versions of the interface and allow users to choose which layer (or version) best suits his or her needs at a given point in time. Layered interfaces have since been applied to a simple text editing application [30], a medical imaging application [29, 29], an operating system [28], and an information visualization application [98], but evaluations have been either informal or have been conducted only with relatively simple interfaces. As a result, several design challenges remain for layered interfaces, including the difficulty of defining layers in a way that meets the needs of most users, and how to support efficient transition between layers [106].

Multiple interfaces [85] is another approach related to layered interfaces. The user can choose to work in the full interface to the application or a "personal" layer, and users specify which individual menu and toolbar items appear in the personal layer. McGrenere, Baecker and Booth [85, 86] conducted an evaluation of a multiple interfaces design for Microsoft Word 2000, comparing it to the application's native adaptive menus. Out of 20 participants, the majority customized their personal layer effectively, and preferred the multiple interfaces design to both the full interface alone and the native adaptive interface.

Split menus, as described in Section 2.3.2, have also been extended to be adaptable. To summarize, Findlater and McGrenere [38] compared adaptive, adaptable, and static split menus and showed that the adaptable split menus were preferred to both static and adaptive split menus. When users chose to customize, the adaptable menus were also faster than the adaptive menus.

Finally, User Interface Façades is an extremely flexible adaptable approach introduced by Stuer-

zlinger, Chapuis, Phillips and Roussel [116]. Façades can be used on top of any existing application and allow the user to reconstruct the interface by duplicating, excluding, and replacing items. The technique has not yet been evaluated, so it is unclear whether users would find it easy to use and effective.

2.3.4 Mixed-Initiative Approaches

Horvitz has identified several principles for the design of mixed-initiative systems that address how to best merge direct manipulation with interface agents [63]. The goal is to incorporate user direction into intelligent agent systems to resolve ambiguities about the user's goals and focus of attention. The Microsoft Windows XP Start Menu, which combines an adaptively created shortlist of programs with the ability for the user to "pin" programs to this list, offers a familiar example of a mixed-initiative design. The interpretation of mixed-initiative user interfaces can vary widely, however, and since this type of interface is not the main focus of this thesis, we give only a few examples here.

One application example was introduced by Thomas and Krogseter, who extended the user interface of a common spreadsheet application [121]. An evaluation with 13 users showed that an adaptive component which suggests potentially beneficial adaptations to the user could motivate users to adapt their interface [90]. More recently, Bunt, Conati and McGrenere [20] showed through an evaluation with 12 participants that providing adaptive suggestions to support customization can be preferred by users over independent customization, and has the potential to decrease customization time. Finally, another earlier example is the adaptive bar, introduced by Debevc et al. [35]. The system adaptively proposes additions or deletions of items on the Microsoft Word toolbar to help users customize, and an evaluation with 16 participants suggested that the adaptive prompting helped users more efficiently build their toolbar.

2.3.5 Modelling Performance

Theoretical assessments of the effectiveness of personalization approaches have been conducted, although they are not as common as user evaluations. Warren [126] used the Model Human Processor to predict the cost or benefit of using an adaptive split menu over a static split menu in a diagnosis system for physicians. Results from applying the model showed that the adaptive system was beneficial in theory, but their model assumed that the user does not have enough familiarity with the menu to anticipate item locations.

More recently, Cockburn, Gutwin and Greenberg developed a model of menu performance that applies to both static and adaptive designs [31]. The model incorporates Hick-Hyman Law and Fitts' Law, and takes stability of an adaptive menu design into account. Results showed the model could accurately predict performance for four types of menus: frequency-based and recency-based split

menus, traditional menus, and morphing menus (where items are resized according to frequency). Empirical results showed that the predicted performance matched actual user performance well for adaptive split menus and morphing menus (where items grow and shrink in size), and could be useful for predicting the overall benefit of novel adaptive menu designs. We note, however, that it is not applicable in its current form to the interfaces studied in this thesis because: (1) it does not take into account whether users will choose to utilize the adaptive predictions or not (as shown in [124] and the Screen Size Study in Chapter 5); and (2) it does not take into account the impact that incidental learning (i.e., awareness) of unused features can have on performance (which we show in Chapter 6).

Although it is not meant to predict performance with personalized menus, Hui, Partridge and Boutilier [65] developed a model to estimate the degree to which an adaptive change to a pull-down menu will disrupt the user's knowledge of the spatial layout of an interface. Based on this model, they also proposed a new adaptive approach that only executes a change when the estimated cost is less than the estimated benefit in the longer-term. In an evaluation with 8 participants, however, the adaptive approach did not offer a performance benefit over static menus, suggesting that further refinement of the underlying model is needed.

In contrast to modelling performance with adaptive interfaces, there has been even less work with adaptable and mixed-initiative interfaces. Bunt, Conati and McGrenere [19] used GOMS modelling to study when and how users should customize their interface to derive the best efficiency gain. The theoretical analysis showed that customization can result in significant performance benefits and that customizing up front before starting a task is more efficient than incrementally customizing the interface as required.

2.4 Designing for Learnability

A goal of some personalization approaches, such as layered interfaces [106] and training wheels [23], is to make it easier for novice users to accomplish their tasks; this is related to learnability. Our measure of awareness is also related to learnability: in a recent evaluation of methods for assessing learnability, Grossman, Fitzmaurice and Attar [59] identified awareness of functionality as one of five categories of learnability issues. Their work was published after we introduced awareness as an evaluation measure [39] for personalized GUIs, but it emphasizes the importance of awareness as a component of learnability.

Although there is a long history of work on learnability, there is notably a lack of consensus on the definition of learnability and how to evaluate it [59]. Many general GUI design guidelines implicitly address learnability, for example, by stressing the importance of semantically relevant labels for features [46, 112] and the need to make interfaces explorable [78]. Here, we provide a brief summary of how people learn to use software applications and examples of typical approaches that

have been developed to explicitly support users in this process, particularly outside of a classroom setting.

Although learning resources such as manuals, online help, and knowledgeable colleagues may be available, users tend to learn through experience, by task-directed exploration of the interface [100]. While this self-directed learning may be effective to a point, one downside is what Carroll and Rosson [24] term the *production paradox*: a user's primary goal is to accomplish his or her task, and they are unlikely to spend much time learning how to use the system beyond this basic goal. The result is that users often asymptote at a level of expertise that allows them to complete their tasks, but they may not have learned the most strategic, efficient methods for doing so. This is reflected in more recent research showing that users prefer just-in-time learning to accomplish their tasks, rather than investing time in advance [100].

Based on a survey of learnability research, Grossman, Fitzmaurice and Attar highlight the distinction between *initial* learnability of an interface and *extended* learnability over time. To improve initial learnability, scaffolding and layered learning approaches are designed to guide novice users through initial learning stages in a training context, by preventing errors when completing specific tasks (e.g., [71]) or providing less detailed instructional material and more functionality over time (e.g., [78, 99, 103]). Reduced-functionality personalization approaches, such as the training wheels approach described in Section 2.3.3 can also be found in a learning context (e.g., [64, 103, 114]), but with added direction from an instructor or course material.

Increasing users' expertise with the system over time is also an important goal of learnability research, since experience alone may not be enough for users to adopt efficient interaction methods [8, 102]. Researchers have explored how to encourage users to adopt more efficient strategies for accomplishing tasks. Bhavnani, Reif and John [9] developed training methods for an instructor-led classroom context to encourage users to develop more efficient interaction strategies, for example, to eliminate repetitive tasks. When teaching users about new features that would help them complete a current task, Lee and Barnard [77] showed that demonstrating the benefit of the new feature at a conceptual level rather than simply introducing the feature as a solution to the current task problem resulted in more use of the new feature on subsequent transfer tasks.

Another approach to improving the user's expertise is to make interfaces more effortful, to encourage learning of more efficient strategies. For example, if there are multiple methods for accomplishing a task, a designer may purposely introduce a time delay to make one method more inefficient with the goal of motivating users to adopt another method. This approach has not been widely explored, but shows potential for learning of keyboard shortcuts [58] and the spatial location of interface elements [32].

2.5 Summary

Our survey of related work highlights the complexity of providing effective personalization mechanisms and identifies several gaps in previous research. First, to our knowledge previous research has not studied the potential impact of GUI personalization on the user's breadth of experience with the interface, that is, the ability to learn about the full set of available features. This issue is relevant to Chapters 3 to 6. Second, despite the relative potential benefit of spatially adaptive GUIs in contexts where navigation is especially time consuming, such as with small screen devices, evaluations have focused on traditional applications and have yielded conflicting results; we address this gap in Chapter 5. Finally, there has been little work on graphical adaptation, and even less on temporal adaptation, a gap which we address in Chapter 7.

Chapter 3

Interview Study: Evaluating a Role-Based Personalization Approach

In this chapter we describe an exploratory interview study with 14 users of a complex development environment called Rational Application Developer (RAD). RAD provides a role-based approach to personalization and the goal of the study was to identify potential benefits and research challenges of approaches to personalization that reduce functionality. This study provides preliminary findings that motivate the more substantial studies presented later in the dissertation (Chapters 4, 5, 6 and 7).

3.1 Introduction and Motivation

Research on coarse-grained approaches to personalization, such as layered interfaces [106], has been limited to relatively simple applications or personalization models [29, 30, 98, 106]. A coarse-grained approach allows large groups of features to be enabled or disabled at once; in contrast, a fine-grained approach enables or disables individual features, as is done with Microsoft Office 2003's adaptive menus, or with multiple interfaces [85]. Since lack of time and difficulty are among the factors that inhibit user-controlled personalization [81], coarse-grained approaches have the potential to provide the benefits of personalization while reducing the burden on the user. However, due to the lack of evaluation of such approaches, we do not fully understand their effectiveness.

The role-based personalization model found in IBM Rational Application Developer 6.0 (RAD) is an example of a coarse-grained approach for a complex, feature-rich application. This approach, shown in Figure 3.1, allows the user to select from a set of user roles, such as Java Developer and Web Developer, and only features associated with those roles are enabled in the user interface. Although CSCW (computer supported cooperative work) applications have on occasion provided user roles to support collaboration, and roles have been proposed to support task-based context switching for the user [108], the research literature does not contain examples of using roles to filter features

in complex user interfaces. An additional difference from the CSCW examples is that RAD's personalization model offers flexibility through multiple levels of granularity, unlike the restrictive definitions of roles that have been found to be problematic in CSCW [54, 109]. Groupware applications that provide user roles also usually do so by providing a small number of distinct interfaces, such as student and teacher interfaces (e.g., [2]), in comparison to the 11 user roles in RAD.



Figure 3.1: Screenshot of RAD's Welcome screen, with an enlargement of the mechanism to change user role shown on the right. A short description is presented for each role on mouseover.

To address the limitations discussed above, we conducted an interview study with 14 users of RAD. This work was conducted while the author was on internship at IBM, and the primary motivation was to provide a preliminary assessment of the role-based personalization mechanism in RAD, with the goal of impacting the new product cycle. However, many of the findings are applicable to coarse-grained personalization approaches in general, and we summarize them here. We particularly focus on feedback regarding partitioning of features, presentation, and individual differences. These issues should be considered by designers of reduced-functionality systems, and offer potentially fruitful areas for further research.

3.2 IBM Rational Application Developer

RAD extends and inherits all user interface components from Eclipse [37], a popular IDE. Shown in Figure 3.2, the key graphical user interface (GUI) personalization components of RAD are as follows:

1. *Workspaces* hold one or more development projects. Users can create more than one workspace,

but can only work in a single workspace at a time. Personalization changes are only persistent within a workspace.

2. *Perspectives* group features by task (e.g., Debug Perspective). The user controls which menu and toolbar items as well as views on the code appear in a perspective, and can also control switching between perspectives. There is often feature overlap between perspectives.
3. *Capabilities* are groups of features that correspond to user tasks on a higher level than perspectives. The features associated with a capability can range from entire perspectives to individual menu and toolbar items within a perspective. When a capability is disabled, the features associated with it are no longer visible. For example, enabling the Java Development capability enables features for creating and testing Java projects, such as a Java-specific text editor, and a menu item to create a new class.
4. *Roles* are groups of capabilities that are potentially overlapping. RAD provides 11 roles on a Welcome screen when the user creates a new workspace. By default, 2 roles are enabled (Java Developer and Web Developer), but the user can disable these and/or enable additional roles. When the user enables a role, this enables the set of capabilities associated with that role; in turn, the specific interface elements associated with those capabilities are made available in the interface. For example, enabling the Tester role will enable 3 capabilities: Core Testing Support, Probekit, and Profiling and Logging.

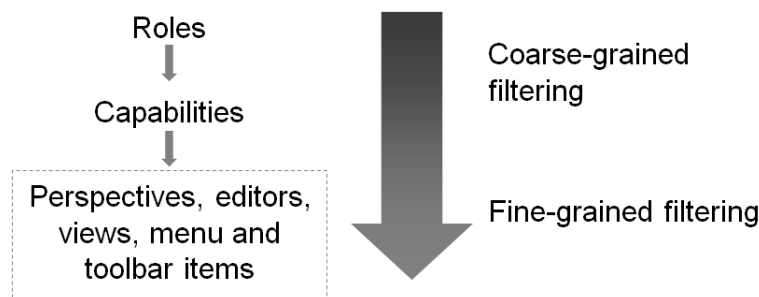


Figure 3.2: GUI personalization mechanisms in RAD. Specific settings at each level are associated with a workspace.

Roles determine a base set of features to include in the interface, and, as the user works, additional features can be exposed or hidden by manipulating capabilities. This can be done both manually, through a user preference dialog that lists all available capabilities, or automatically, through trigger points in the interface. Trigger points offer a small amount of adaptive prompting in an otherwise adaptable personalization model: for example, when creating a new project the user can choose to show all types of possible projects; if the new project is associated with a disabled capability, the system will prompt the user to enable that capability.

3.3 Interview Methodology

Each interview was 1 hour long, with 32 semi-structured questions to understand use of roles and capabilities, and overall personalization practice (see Appendix A). At the end of the interview there was a debriefing and unstructured discussion period on managing user interface complexity. All interviews were conducted by the same researcher and were recorded, transcribed and coded for analysis. Since the interviews were exploratory, we did not set hypotheses beforehand. Instead, we used an open coding technique to develop categories of data [115]. This iterative process allowed us to identify emergent themes, and confirmed some of the focus areas of our investigation. We separated pure usability issues from what we consider to be the more generalizable benefits and challenges of reducing functionality. Almost all questions were open ended and participants were encouraged to speak freely, so the number of people who mentioned a point should be considered a minimum.

Through developer mailing lists and word of mouth, we recruited and interviewed 14 professional software developers (11 male, 3 female). They had between 2 and 30 years of software development experience ($M = 11$, $SD = 9$) and reported spending over 30 hours per week using an Eclipse-based development platform ($SD = 13$). All participants had experience with RAD, but the amount varied from less than a month for 3 participants to 12 months for another participant ($M = 4.1$, $SD = 3.2$). This was representative of the user base, since RAD had only been released 6 months before we conducted the study; the participant with 12 months of experience had initially used a pre-release version. Participants reported using RAD to develop a variety of applications, including: Web (7 participants), J2EE (4 participants), Java or plug-ins for Eclipse (6 participants), and database (1 participant). Three participants used Eclipse as their primary IDE, rather than RAD, and some questions pertaining to functionality found exclusively in RAD were not asked of these users (noted when applicable in the next section). Participants were compensated for their time with a gift worth approximately \$5.

3.4 Findings

We first briefly discuss overall personalization practice to provide context for the findings on roles and capabilities.

3.4.1 Overall Personalization Practice

RAD provides 11 perspectives by default, though users can increase this by saving personalized perspectives and installing additional plug-ins. On average, participants made use of 4 to 5 perspectives. Most participants (11) had multiple workspaces, with the median being 2 to 3 workspaces. All participants generally made at least minor personalization changes to each workspace, including

opening and closing different views on the code, changing the layouts of perspectives, and changing code formatting preferences, but none of the participants personalized their menus and toolbars individually. A reset feature is provided for perspectives, and 6 participants reported occasional use of this feature when they had changed their perspective significantly. Users can also create new perspectives by first personalizing a perspective, then saving it under a new name. Only 1 participant used this feature.

3.4.2 Challenges in Coarse-Grained Personalization

As expected based on our participants' varied exposure to RAD, we found that people had different degrees of understanding about how roles and capabilities technically worked. While almost all participants (12) were aware of capabilities, only 8 of the 11 participants who did not use Eclipse as their main development platform were aware of roles, and only 6 of those knew how to change them. Interpretation of results should be made in this context.

The majority of participants (8) explicitly stated they liked roles or capabilities in principle, that is, their potential to personalize the interface by reducing the number of features. When asked if they would remove roles and/or capabilities from the interface, only 1 participant suggested removing both. While this positive response should motivate further work on roles and capabilities, several issues affected the user experience and these can be broadly grouped with respect to partitioning of features, presentation, and individual differences.

Partitioning features

We identified several challenges related to partitioning features into meaningful groups.

Fine-grained capabilities were more popular than coarse-grained roles because they better matched perceived needs. While roles and capabilities both offer high-level feature grouping for adaptable personalization, they do so at different levels of granularity. Participants generally chose to customize their interface by using the finer-grained capabilities rather than using the roles. Part of the reason was that they felt the variation in tasks performed by users nominally in the same work role made it difficult to define roles. We asked all but the 3 participants who used Eclipse as their main IDE which roles they would categorize themselves under, and we compared this to the roles which were actually enabled in the workspace they had accessible during or after the interview. All but 2 people identified with several more roles than were enabled in their workspaces.

Trigger points and capabilities were useful because they allowed the user to enable features as needed rather than predicting needs in advance. Five of the 6 participants who knew how to change roles generally left the default roles when they created a new workspace even though 3 of them had changed their roles at some point in an earlier workspace. They found it easier to enable features automatically through trigger points or by manually enabling capabilities, and 3 of those participants

considered roles to be irrelevant because instead, they could simply change their capabilities. For example, P8 said:

“It’s not very intuitive, saying ‘This is what I’m going to do’ up front.” (P8)

Only 1 participant used roles as his primary method of enabling features. This was not necessarily because the role matched his work practice better than it did for other participants: he stated he had chosen this specific role (Advanced J2EE) because it appeared to be the most comprehensive. Thus, it made it easy to enable a large set of features with a single click.

Partitioning based on task was more effective than partitioning based on expertise. Our analysis also suggests that the criteria by which roles are defined impacts the effectiveness of the personalization model. All 11 of the roles in RAD group features in a task-oriented manner; for example, the Java Developer role is associated with features that are likely to be needed by that type of developer. However, 4 of the roles were also distinguished by expertise level: Web Developer Typical versus Web Developer Advanced and Enterprise Java versus J2EE Developer. The former role in each of these pairings represents only a subset of the features of the latter. Eight participants expressed concern over the difficulty of distinguishing between the expertise-oriented roles. For example, when asked to identify which roles he fits under, P7 said:

“The main ones would be Enterprise Java and Modeling, and I guess the Advanced J2EE. Although I have no idea why there’s Enterprise Java and Advanced J2EE. I almost think it would be better to just have one.” (P7)

Although partitioning by expertise has been shown to be effective for novice users [23], our findings suggest that it may not be as effective for differentiating between the tasks of more experienced users (intermediate vs. expert users).

Presentation

Effective communication of a complex personalization model to the user is non-trivial. From this, several issues of presentation arose.

Capabilities more closely matched concrete tasks, so were easier to interpret. Many participants (8) found it difficult to map from a name or short description of a role or capability to actual features in the interface, thus making it difficult to know how to effectively personalize their interface. For example, P1 expressed this frustration:

“If I need something but if I dont know which capability I need to [enable], how can I use that?” (P1)

While some of this may be attributable to issues with partitioning features, it also highlights the challenge of effectively communicating the personalization model to the user when the model is complex, such as RAD's, and contains multiple levels of granularity. It would be interesting to explore whether communicating the underlying mapping of roles to features more effectively would increase their adoption relative to capabilities.

Designers need to promote the ability to discover unknown or unused features while still filtering what is presented to the user. As discussed at the beginning of Section 3.4.2, the majority of participants liked the potential for roles and capabilities to reduce the number of features in their interface through personalization. However, more than half the participants (8) were concerned about hiding features and not being able to find features when some roles or capabilities were disabled, a finding consistent with previous work with word processor users [87]. Because of this concern, 4 participants mentioned that they generally enabled all features to ensure that they would be able to find what they needed. Although this behaviour may be due to individual differences (see below), it defeats the purpose of having roles and capabilities in the first place. Participant 6 offered a typical quote from this group of participants:

“I go to the preferences dialog and I turn everything on. I hate having stuff hidden.”
(P6)

The concern over hiding features stemmed from both: (1) the need to locate features of which the user is already aware, and (2) the ease with which users can learn about and use new features in the user interface. As an example, Participant 7 expressed frustration when trying to find a new feature that was not enabled in his personalized interface:

“... if it's something new that I haven't done before I don't really know where the menu should be in the first place and if I right click the object and I don't see it I have to go and start searching somewhere else and I don't know that the reason that I don't see it is because I'm not doing the right thing or because I don't have the capability enabled.”
(P7)

In contrast to not knowing where to find a new feature that the user already believed should exist, there was also the issue of learning about more advanced features of which the user was not yet aware:

“I guess any situation where you have two versions of something, a simple one and an advanced one, the advanced is disabled, I think you should be prompting to say there is an advanced one if you want to use it but we haven't enabled it by default.” (P3)

Changing requirements concern users. Our participants identified three situations in which they would be concerned about only having a filtered set of the features in the interface: when their role

evolved, such as from a developer to a manager; when they temporarily needed a set of features associated with another role; and when they wanted to engage in exploratory behaviour of the interface for a short period of time.

Individual differences

Finally, we found that different participants had different reactions to reducing functionality in the user interface. Some felt overwhelmed by having many features while others were not bothered by extra features and preferred not to filter any features. As such, we need to cater to both feature-keen and feature-shy users [87], and to increase system trust, especially for those users who may be reluctant to personalize even when a reduced-functionality interface could be more efficient. Four participants immediately enabled all features when creating a new workspace. To illustrate this, when asked which of the roles she would want enabled, P5's response was: "Every single one of them!" This behaviour supports the inclusion of a toggle mechanism, such as that provided in the multiple interfaces approach [85], to provide quick access to the full feature set for this type of user.

3.4.3 Summary of Design Implications

Participants preferred to use finer-grained capabilities to roles, for several reasons that can inform future designs: (1) capabilities more closely matched the tasks a user performed, while roles were broader, not necessarily matching an individual users tasks; (2) capabilities were more concrete, so it was easier to interpret the mapping from capabilities to individual features; and (3) capabilities could be easily enabled on an as-needed basis. Grouping of features based on advanced expertise levels was also less effective than grouping by task. As well, although most users wanted to filter features in their interface, it is important to consider how easily unknown or unused features can be discovered. Finally, for those users who do not want to filter any features, an easy toggle mechanism enabling the full feature set should be provided.

3.4.4 Limitations of the Study

These interviews were formative and allowed us to identify some of the more critical issues faced by long-term users of a coarse-grained personalization mechanism. However, replication of the protocol with a larger number of users and with a software application in a different domain will be important for understanding the completeness of the set of issues we identified. Particularly, although our participant sample included a range of expertise both with the application and the task domain, all participants were necessarily highly technical. Less technical users may have different expectations for a personalization mechanism and may identify additional challenges with coarse-grained customization.

3.5 Conclusion

This exploratory interview study provided us with a better understanding of issues that arise in long-term use of a personalized interface, and in particular allowed us to identify several open issues in designing coarse-grained adaptable personalization mechanisms. Our findings suggest that smaller, task-oriented groupings of features (i.e., capabilities) may be more effective than role-based groupings. The design implications are especially applicable for role-based and layered interfaces. The challenges we have identified with respect to partitioning of features, presentation, and individual differences highlight potentially critical design choices, and should guide further research in the area.

Of particular importance for shaping the remainder of this dissertation was participants' concern about hiding features and the potential impact that personalization could have on their ability to learn about and use new features (Section 3.4.2). This finding, along with previous research findings on word processor use [87], motivated us to introduce and define awareness as a novel evaluation metric, which we applied to three subsequent studies: the Layered Interface Study (Chapter 4), the Screen Size Study (Chapter 5), and the New Task Study (Chapter 6).

Chapter 4

Layered Interface Study: Measuring Feature Awareness and Core Task Performance

In this chapter we explore the impact that personalization may have on the user's ability to learn about new features, a tension identified in the Interview Study (Chapter 3) and in previous work by McGrenere and Moore [87]. We introduce feature awareness as a new evaluation metric to assess the degree to which users are conscious of the full set of available features in the interface and present the Layered Interface Study, a proof-of-concept study that compared two types of layered interfaces to a static control condition. Results show a measurable trade-off between performance on core tasks and awareness for a layered interface design that provides a minimal set of features.

4.1 Introduction and Motivation

Personalization approaches, such as layered interfaces [106], have been proposed by several researchers to manage user interface complexity by reducing the number of features available to the user. Evaluations of such reduced-functionality interfaces, however, have been limited in number and have focused largely on the benefits. Our goals for the Layered Interface Study were: (1) to introduce awareness as a new evaluation metric; (2) to compare two 2-layer interfaces to a control interface by measuring awareness and core task performance; (3) to show that, in comparison to measuring only core task performance, the inclusion of both awareness and core task performance reveals a more comprehensive understanding of the impact of reducing functionality.

Several methods to reduce functionality through personalization have appeared in the research literature and in commercial applications. The layered interfaces approach, for example, gradually introduces new functionality to the user by starting with a simple interface containing a core set of

features, and allowing the user to control his or her transition to increasingly feature-rich interface layers [106]. In contrast to many personalization methods that enable, disable, or otherwise modify individual menu or toolbar items (e.g., [38, 85]), layered interfaces offer a relatively coarse-grained approach; in this respect they are similar to the role-based personalization studied in Chapter 3. Examples of layered interfaces to date allow the user to transition from between 2 to 8 layers, and evaluation has been mainly qualitative [29, 27, 28, 30, 98, 106].

A related approach to reducing functionality is the multiple interfaces approach, which offers users a “personal” interface in addition to the full interface of the application [85]. The user can easily switch between the two interfaces and specify the menu and toolbar items contained in his personal interface. Another earlier reduced-functionality approach is the training wheels interface, which blocks the use of advanced functionality for novice users but does not remove it from the interface [23]. In contrast to these, adaptive mechanisms can also be used to automatically reduce functionality; for example, Microsoft Office 2003 provides personal menus that contain only an automatically-generated subset of features when they are initially opened. Mixed-initiative mechanisms that combine both system and user control have also been proposed, most commonly through the use of adaptive suggestions to support the user’s customization (e.g., [20]).

Evaluations of these reduced-functionality personalization approaches have shown that they can make novice users faster, more accurate and more satisfied with the interface [23], and that such approaches can be preferred by a large proportion of intermediate and advanced users [85]. However, we argue that satisfaction and initial speed only reflect part of the impact of functionality reduction. When features are removed from the interface, the user’s level of awareness of the full feature set is also affected. A severely feature-reduced interface may promote ease of accessing functions, but likely impedes awareness of those functions only available in the full application interface. A survey of 53 users of Microsoft Word 97 reflects this tension: while many of the users requested that their unused functions be “tucked away”, many also indicated that it was important that they be continually able to discover new functions [87].

To explore the tension between personalizing to improve performance versus the user’s ability to learn about new features, we define feature awareness as a new evaluation measure. Awareness is the degree to which users are conscious of the full set of available features in the application, including those they have never used; it is a measure of the secondary learning that may occur as the user performs a primary task. We propose that measuring awareness in conjunction with performance (efficiency) will be particularly valuable for personalized interfaces. Together, these measures offer a broader understanding of the impact of working in a personalized interface. Although awareness does not impact performance on routine or known tasks (those supported by the personalization and familiar to the user), it has the potential to impact performance on new tasks. We thus distinguish core task performance from new task performance.

We conducted an experiment to empirically validate the trade-off between core task performance

and awareness and to provide the first controlled comparison of more than one multi-layer interface. Our study compared two 2-layer interfaces, Minimal and Marked, to a control condition and showed that core task performance in the Minimal approach was significantly better than in the Control, but that participants were more aware of features in the Control. The Marked approach provided little benefit over the other two.

4.2 Introducing and Defining Awareness

Awareness applies to the full set of features in an application, but to measure it we focus particularly on the subset of features that have not yet been used. In contrast, the focus of personalized interfaces is often to improve core task performance, that is, performance of routine tasks that require more familiar features.

We predicted that lower awareness of unused features in the present would, in turn, impact performance when the user is asked to complete new tasks. We thus differentiated new task performance from core task performance. For new, complex tasks, performance should be impacted both by the time it takes to complete the steps with which the user is already familiar, which are those most likely to be supported by the personalized interface (core task performance), and the time it takes for the user to “discover” how to complete new, unfamiliar steps (new task performance). The time to complete the latter steps would be in part related to prior awareness of not-yet-used features.

We operationalized two measures of awareness:

1. **Recognition rate of unused features.** As a straightforward measure of awareness, we measure the ability of experienced users to recognize features that are available in the interface, but that they have not yet used (see Section 4.3.6 for more detail on the format of our recognition test).
2. **New task performance.** The speed with which experienced users can locate, when prompted, features that they have not yet used. In contrast to the recognition rate, this is an applied measure of awareness and requires more effort to assess because it requires asking users to perform new tasks. However, it provides different knowledge about the feature and is a more direct indicator of the impact that awareness may have in the longer term.

Awareness is only one component of performance when selecting graphical user interface elements. Performance and user satisfaction also depend on a number of factors, including user characteristics, such as experience or cognitive and motor abilities, and interface characteristics, such as layout. Awareness is one aspect of the user’s experience that is particularly important for personalized approaches, where the impact on awareness may be greater than in more traditional user interface designs. Since personalization approaches will impact core task performance and awareness to differing degrees, distinguishing between the two measures allows for a more nuanced

comparison of designs than measuring performance alone. For more detailed discussion see Chapter 6.

4.3 Experimental Methodology

This study compared two 2-layer interfaces to a control condition (a default full interface) based on core task performance and awareness. We chose Microsoft PowerPoint 2003 as our experimental application. Though not as complex as some applications (e.g., animation software), PowerPoint does not require specialized domain expertise, easing participant recruitment, and the menus and toolbars are highly programmable.

4.3.1 Interviews to Define Command Sets

To inform the design of the experimental conditions and tasks, we interviewed 10 frequent users of PowerPoint XP and 2003 from varying backgrounds (academic, business and medical). For each of the 361 selectable items (commands) found in the pull-down menus and on the 12 main toolbars, we asked users to specify their frequency of use (never, irregular or regular). From this, we defined a baseline interface, composed of menus and toolbars that were used at least irregularly by at least half of the 10 users. This included all the menus and the Standard, Formatting, Drawing and Picture toolbars, a total of 260 commands; it did not include several context-based toolbars that are not visible by default. (Two duplicate commands were also removed: Slide Show appeared twice in the menus, and Font Color appeared twice in the toolbars.)

We then categorized the commands in our baseline interface according to two independent dimensions: (1) *basic* commands are at least irregularly used by at least 8/10 of our users, while the remaining are *advanced*; and (2) *MSOffice* commands are common to the default layout of other MS Office or MS Windows applications (such as Save) while *PPT* commands are not found in other MS Office applications and are considered specific to PowerPoint. These categorizations impacted our interface and task designs, and we often refer to the intersections of the sets, for example, *MSOffice-basic* commands. The relative frequencies are shown in Table 4.1. We note that our data and categorization offer only an approximation of PowerPoint commands and their usage. This is sufficient for our experimental setup, but more accurate usage data and categorization adjustment would be needed for a deployable reduced interface for PowerPoint.

4.3.2 Conditions

We evaluated three conditions: two 2-layer conditions (Minimal and Marked), and a control condition (Control). In the layered conditions, participants completed a simple task in the initial interface layer of the respective condition, then transitioned to a full interface layer for a second, more com-

	Basic	Advanced	Total
PPT	12 (5%)	123 (47%)	135 (52%)
Msoffice	32 (12%)	93 (36%)	125 (48%)
Total	44 (17%)	216 (83%)	260 (100%)

Table 4.1: Breakdown of baseline command set (from the menu bar and the Standard, Formatting, and Picture toolbars).

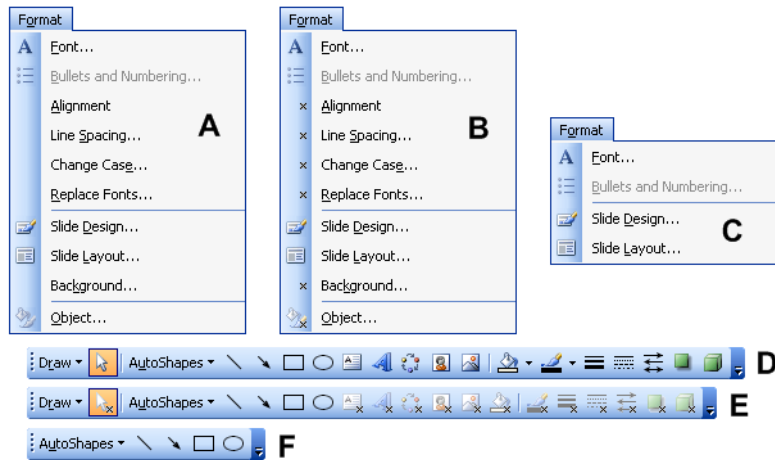


Figure 4.1: Sample menus and toolbars from the three experimental conditions: *A*, *B*, and *C* show the Format menu for the full, marked and minimal layers, respectively; *D*, *E*, and *F* show the Drawing toolbar for the full, marked and minimal layers, respectively. (The marked toolbar is narrower than the full one because the drop-down arrows on some commands could not be replicated for blocked functions.)

plex task. This simulated the predicted use of a layered interface, where users start in a reduced layer to complete easier tasks, then transition to more complex layers for advanced tasks [106]. In the Control condition, participants completed both the simple and complex task in the full interface layer. Each of the layers is described below and Figure 4.1 shows samples of their menus and toolbars. Context menus and keyboard shortcuts were disabled in all conditions to constrain the interaction and to focus on the persistent visual complexity of the menus and toolbars. The MS Office adaptive menus were also turned off.

- **Full interface layer:** The baseline interface, which contained 260 commands, as described above.
- **Minimal interface layer:** Contained only the 44 basic commands (both MSOffice and PPT). Since the Tools menu, Window menu, and Picture toolbar contained no basic commands they did not appear in this layer.
- **Marked interface layer:** Extended Carroll and Carrithers' training wheels approach [23] by

visually marking as well as blocking access to all advanced commands, leaving only the 44 basic commands accessible. Marking was achieved by fading a command's icon (if it had one) and adding a small 'x' (see Figure 4.1, B and E); if selected, a dialog box informed the user that "This item is unavailable for this task". Limitations in PowerPoint's application programming interface (API) forced two secondary design decisions: (1) submenus with all blocked commands were completely removed, and their parent item was visually marked, which reduced the total command set to 210; and (2) only the icon was changed on blocked menu items (ideally the APIs would have allowed us to pilot test options for changing the background or text colour as well).

The Minimal and Marked conditions both provided a small feature set which would increase over time in real usage. A personalization approach that moves from less to more functionality should maximize core task performance in the initial interface states, but the awareness of potentially useful advanced functionality is likely to be compromised. Comparing Minimal to Marked provided insight into the impact of different types of visibility of blocked functions: we anticipated that visually distinguishing, but not completely removing blocked functions could offer a compromise between core task performance and awareness.

4.3.3 Task

We designed two tasks (simple and complex), each consisting of a list of step-by-step instructions to modify a pre-existing short presentation (see Appendix B). On each step, the user performed one interface operation: steps requiring PPT-basic or PPT-advanced commands were interspersed among MSOffice-basic commands, navigation, and text entry steps to create a realistic task flow. The instruction indicated when to use a menu or toolbar, but did not specify the exact name of the command. For example, Figure 4.2 shows a screenshot of the experimental system with an instruction that specifies that the participant should use a menu command to show the slide design options (which maps to Slide Design command in the Format menu).

The command distribution by task is shown in Figure 4.3. The tasks were as follows:

- **Simple task:** This relatively short task could be completed in any of the three interface layers. It included all 12 PPT-basic commands, 6 of which were repeated twice to increase task length, making it more realistic and allowing participants to develop some familiarity with the interface (18 PPT-basic invocations in total); 12 MSOffice-basic commands were also included.
- **Complex task:** This longer task could only be completed in the full interface layer, and introduced advanced functionality, such as animation. It included 18 PPT-advanced commands in addition to the exact same set of commands used in the simple task.

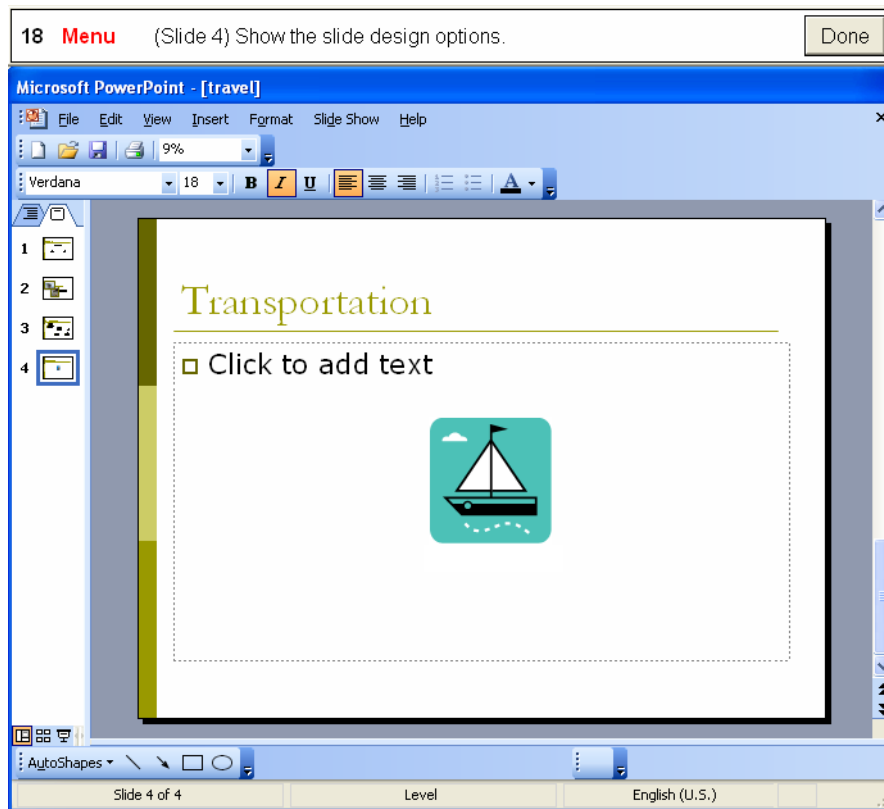


Figure 4.2: Screenshot of experimental system in minimal interface layer; at the top of the screen, the current instruction specifies that a menu item is required.

4.3.4 Design, Participants and Apparatus

A between-subjects single factor (interface) design was chosen to prevent any learning confounds. Thirty participants (17 females) between the ages of 19-55 were randomly assigned to each of the three conditions (10 per condition). Participants were students and community members recruited through campus advertising and a local participant pool. They were screened so that they had either never used PowerPoint or had at most infrequently loaded and viewed a presentation created by others. Each participant was provided with \$20 to defray any costs associated with their participation.

The experiment used a 1.1 GHz Pentium PC with 640 MB of RAM, 18" LCD monitor, and running MS Windows XP and Office 2003. The experimental versions of PowerPoint, one of which is shown in Figure 4.2, were coded in Visual Basic for Applications 6.0. Instructions were given one at a time at the top of the screen. When the participant believed she had correctly completed a step, she clicked the "Done" button and the next instruction appeared. The system recorded all timing data.

critical to completing subsequent steps, the experimenter corrected the situation.

4.3.6 Measures

Each step was measured as the time elapsed until the user clicked the “Done” button. All questionnaires and the recognition test are found in Appendix B.

Core task performance

Core task performance was measured as the time to complete the 18 PPT-basic steps. We did not analyze performance on the MSOffice-basic steps since variation in previous MS Office experience would have confounded the results.

As a secondary measure of core task performance, we also defined *transfer performance* as the time to access familiar features after the user transferred from one layer or version of the interface to another one. We measured this as the time to complete the 18 PPT-basic steps in the complex task.

Awareness

We measured awareness using the two methods outlined in Section 4.2:

1. **Recognition rate of unused features:** To test the recognition rate of unused features, we used a questionnaire listing 20 of the PPT-advanced functions that were present in the full interface layer but were not used for either of the two tasks, and five distractor functions (commands that do not exist in PowerPoint, but could be believed by a novice to exist; e.g., Assign Slide Manager). Icons were also provided for those commands that had one. Half of the valid commands were menu items and half were toolbar items. The distribution of commands tested is shown with a ‘●’ in Figure 4.3.

For each item, participants checked one of three options: (1) Yes, I definitely recall seeing this item; (2) I vaguely recall seeing this item; (3) No, I didn’t see this item. Based on the number of items that were marked as definitely recalled, we then calculated the *corrected recognition rate*, a commonly-applied method in psychology to account for individual variation in the amount of caution a participant applies when responding to a memory test; it is simply the percentage of targets correctly remembered minus the percentage of distractors incorrectly chosen [5]. When an individual user’s corrected score was negative, we assigned him/her a score of zero.

2. **New task performance:** We used time to complete the 18 PPT-advanced commands in the complex task as an indirect measure of the impact awareness can have on performance. These

commands were used for the first time in the second task, which all participants completed using the full interface layer. A difference in access time for these commands should be a result of different levels of awareness gained during the simple task.

Secondary objective measures

Our secondary objective measures included timeouts, errors, and exploration. A timeout occurred if the participant was unable to complete the task step within 2 minutes. Errors only included incorrectly completed steps not already counted in timeouts. Exploration was defined as the number of toolbar or menu items that a participant selected before selecting the target item, or in the case of incorrectly-completed steps, before clicking “Done.”

Subjective measures

We report six subjective measures (Appendix B contains the full questionnaires). After each task, all participants ranked on a 5-point Likert scale the degree to which they felt overwhelmed by the amount of “stuff” in the menus and toolbars, and how difficult it was to navigate through them. Additionally, after completing the second task, Minimal and Marked participants ranked on a 5-point Likert scale how easy they found it to transition from the menus and toolbars in the first task to those in the second, and whether they preferred those in the first task to those in the second. In follow-up interviews, these participants were also asked which version they would prefer to use in the future, and whether or not they could see themselves switching between the two versions for different tasks.

4.3.7 Hypotheses

Our main hypotheses were as follows:

- H1.** Core task performance: Minimal is faster than Marked, and Marked is faster than Control. Related work has shown that novice users were faster with a training wheels version of an interface than the full interface [23].
- H2.** Transfer performance: No significant difference between conditions. This hypothesis is also based on previous training wheels research, showing that users who initially used the training wheels interface performed no differently on a follow-up similar task in the full interface than those who had used the full interface from the outset [25].
- H3.** Recognition rate of unused features and new task performance: Control is better than Marked, and Marked is better than Minimal. To differing degrees, Minimal and Marked reduce the amount of interaction with advanced features and should result in lower awareness of those features.

4.4 Results

We performed one-way ANOVAs on each dependent measure, except where noted. All pairwise comparisons were done with t-tests protected against Type I error using a Bonferroni adjustment. On average across all conditions, the simple task took 15.5 minutes ($SD = 6.3$) while the complex task took 26.1 minutes ($SD = 5.3$).

In all subsequent chapters, statistical tests were done with SPSS statistical software versions 13-17. As well, pairwise comparisons based on ANOVA results were run as posthoc tests with a Bonferroni adjustment using SPSS's General Linear Model. We report measures which were significant ($p < .05$) or represent a possible trend ($p < .10$). Along with statistical significance, we report partial eta-squared (η^2), a measure of effect size, which is often more informative than statistical significance in applied human-computer interaction research [74]. To interpret this value, .01 is a small effect size, .06 is medium, and .14 is large [33].

4.4.1 Core Task Performance

As expected, the conditions did impact core task performance significantly differently ($F_{2,27} = 4.03, p = .029, \eta^2 = .230$). The means and 95% confidence intervals are shown in Figure 4.4. Pairwise comparisons showed that Minimal participants were significantly faster on this measure than Control participants ($p = .027$), but no other comparisons were significant.

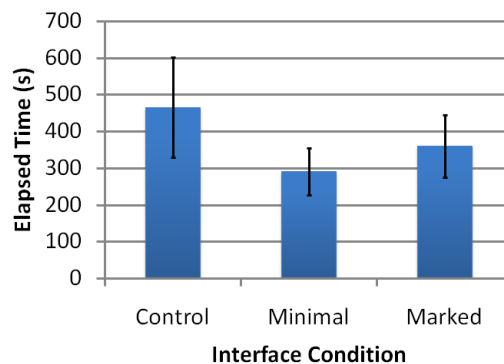


Figure 4.4: Mean core task performance, measured using the PPT-basic steps in the simple task ($N = 30$). Error bars show 95% confidence intervals.

Also as expected, no significant effect of interface was found on transfer performance ($F_{2,27} = .708, p = .501, \eta^2 = .050$). The overall average to complete the PPT-basic steps in the complex task was 211 seconds ($SD = 68$).

For completeness, we ran an ANOVA on the MSOffice-basic steps in both tasks, and, not surprisingly, found no significant differences. This suggests that previous experience with MS Office dominated over interface condition for these steps.

	Correct targets (%)	Incorrect distractors (%)	Corrected recognition (%)
Control	46(<i>SD</i> = 16)	10(<i>SD</i> = 17)	36(<i>SD</i> = 16)
Minimal	36(<i>SD</i> = 18)	24(<i>SD</i> = 30)	16(<i>SD</i> = 15)
Marked	43(<i>SD</i> = 19)	12(<i>SD</i> = 14)	31(<i>SD</i> = 14)

Table 4.2: Average awareness scores as percentage of items answered affirmatively (N = 30).

4.4.2 Awareness

For the recognition test, there was a significant main effect of condition on corrected recognition rate ($F_{(2,27)} = 4.81, p = .016, \eta^2 = .263$). Comparing the recognition rates with the core task performance results in the previous section, better core task performance generally matched with low awareness and vice versa. Control participants had an average corrected recognition rate of 7.1 out of 20 items ($SD = 3.2$), which was significantly more than Minimal participants ($p = .019$), who only remembered on average 3.1 items ($SD = 3.0$). A trend suggested that Marked participants, scoring 6.2 ($SD = 2.9$) on average, were aware of more than Minimal participants ($p = .090$). Table 4.2 shows the hit rate, false alarm (distractor) rate and corrected recognition rate from the awareness recognition test; the difference of about 20% between the Control and Minimal participants for corrected recognition rate represents a difference of 4 items on the recognition test.

Unexpectedly, no significant effect of interface condition was found on new task performance ($F_{2,27} = .172, p = .843, \eta^2 = .013$). Across all conditions, the PPT-advanced steps took on average 780 seconds ($SD = 167$). The means are shown in Figure 4.5.

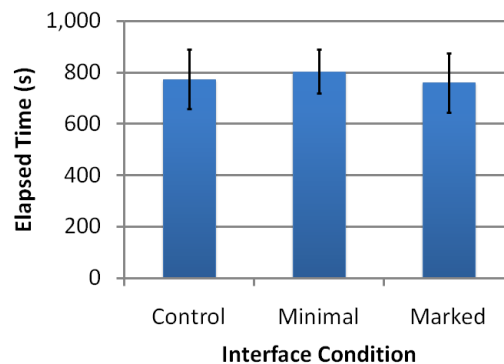


Figure 4.5: Mean new task performance as measured using the PPT-advanced steps in the complex task (N = 30). Error bars show 95% confidence intervals.

4.4.3 Timeouts, Exploration, and Errors

A timeout occurred whenever a participant failed to complete a task step within 2 minutes. Minimal participants never timed out in the simple task, in comparison to Control participants, who had 1.6 timeouts on average ($SD = 2.1$), and Marked participants, who had 0.4 timeouts on average ($SD = 0.7$). Although a non-parametric Friedman test did not show a significant main effect of condition on timeouts, the means suggest that our core task performance measures favoring Minimal were conservative, since Control participants' scores would have been worse without timeouts than they were with timeouts. In the complex task there were more timeouts (across conditions: $M = 2.3, SD = 1.6$), but not surprisingly no significant effect of condition was found.

There was a main effect of condition on exploration in the simple task, measured as the number of times the participant selected a menu or toolbar item that was not the target item for that step ($F_{2,27} = 4.79, p = .017, \eta^2 = .262$). Control participants selected on average 18.6 items ($SD = 13.3$) while exploring, which pairwise comparisons showed to be significantly more than the average of 6.9 ($SD = 6.5$) for participants in the Marked condition ($p = .023$). A trend also suggested that Control participants explored more than the 8.7 average ($SD = 5.4$) of Minimal ($p = .066$). In the complex task, no significant differences were found.

Mean error rates, defined as the number of incorrectly completed steps that were not already counted as timeouts, were uniformly low (simple task: $M = 1.8, SD = 1.6$; complex task: $M = 1.7, SD = 1.6$), and no significant differences were found. The experimenter stepped in on average 0.9 times per participant to correct errors that would have affected further steps ($SD = 1.3$).

4.4.4 Subjective Responses: Questionnaires and Interviews

There was a significant main effect of condition on the Likert scale responses for the degree to which participants felt overwhelmed by how much “stuff” there was in the menus and toolbars after the first task ($F_{2,27} = 4.50, p = .021, \eta^2 = .250$). Pairwise comparisons showed that Marked participants felt more overwhelmed than Minimal participants ($p = .020$). In terms of navigation, a trend also suggested that interface condition may have had an effect on the degree to which participants felt it was difficult to navigate through the toolbars in the complex task ($F_{2,27} = 2.54, p = .098, \eta^2 = .158$).

Using one-tailed t-tests, we evaluated the two Likert-scale questions that were asked of only Minimal and Marked participants. Marked participants felt more strongly than Minimal participants that they were easily able to transition between the menus and toolbars used in the two tasks ($t_{18} = 1.89, p = .038$). Minimal participants preferred their initial interface to the full interface more than Marked participants did ($t_{18} = -2.76, p = .007$).

During interviews, participants were asked which interface they would prefer to continue using: the reduced-functionality layer or the full interface layer. Minimal participants overwhelmingly chose the minimal layer over the full layer (9/10 participants) while only 1 out of 10 Marked par-

ticipants chose the marked layer over the full layer. This replicates recent subjective findings of training wheels on a modern word processor [6]. Trends suggested that participants who had used the minimal interface could see themselves switching between a minimal and full layer for different tasks (8/10 participants), whereas participants who used the marked interface felt exactly the opposite about their interface (8/10 participants). Note that because of low power no statistically significant results were found using Fisher's exact tests on the preference results within each interface condition.

4.4.5 Summary

We summarize our findings in terms of our hypotheses:

- H1.** *Core task performance. Partially supported:* Minimal had significantly better core task performance than Control, but there were no other significant differences.
- H2.** *Transfer performance. Supported:* No difference for transfer performance.
- H3.** *Awareness. Partially supported:* In terms of recognition rate, Control was better than Minimal and a trend suggested that Marked was better than Minimal. There were no other significant differences for the recognition test, nor any for new task performance.

4.5 Discussion and Future Work

Awareness measure adds value. The comparison of the Control condition to the Minimal 2-layer condition showed that there is a measurable trade-off between core task performance and awareness. Taken in isolation, the core task performance results replicated related work on training wheels interfaces [23], and could lead us to reach the straightforward conclusion that a minimal 2-layer interface is better than the full interface alone: it was faster in the first task and had no cost when transferring to the full interface layer for the second task. By teasing apart performance and demonstrating that improved core task performance can come at a cost of decreased awareness, we provide a richer understanding of the experience.

Two-layered minimal interface is promising. The qualitative and quantitative data together suggest that a two-layer minimal interface offers significant advantages over a default full functionality interface alone, although the findings need to be tempered by its effect on awareness. Eight out of 10 participants indicated that they would prefer to have a minimal interface in conjunction with a full interface. Core task performance was better in the simple task in the Minimal condition, and transfer performance in the complex task was no worse than in the Control condition. The Control condition did, however, result in higher awareness recognition test scores. We speculate that if users could freely switch between a minimal and full interface, the impact on awareness could be

smaller, but further research is required to substantiate that claim. Also, our study protocol did not have users interacting with the personalization mechanism since the experimenter set the interface layer for each task. Although the goal of reducing functionality is to reduce complexity, the very inclusion of a mechanism to do so adds some complexity to the interface. This impact needs to be outweighed by the beneficial effects of the personalization.

Two-layered marked interface is problematic. We theorized that visually marking, yet not removing, blocked functionality would offer a compromise between core task performance and awareness. This was not shown in our study. A trend-level result suggested the Marked condition resulted in higher awareness recognition test scores than the Minimal condition. Combining this with the means in Figure 4.4 suggests that the Marked condition may have a small positive effect on core task performance and awareness but we simply did not have sufficient power to detect this. However, the preference of 9 out of 10 participants for the full interface over the marked one is a strong indicator that even if a 2-layer marked interface offers a small performance improvement over a full interface, it would not achieve widespread adoption. The marked interface, in its current form, is not a fruitful direction for further research. An alternative would be to try a different form of visual marking that allows users to filter out advanced items more easily, which could lessen the negative reaction. Although not in the context of layered interfaces, we study two alternative forms of visual marking (colour highlighting and ephemeral adaptation) in Chapter 7.

Measuring awareness is challenging. The two measures of awareness in our study produced inconsistent results. The recognition test of unused features partially supported our hypotheses that the Minimal condition would have the least awareness, the Control the most, and the Marked condition would be in between. By contrast, the second measure of awareness, new task performance, provided no support for our hypotheses. This is likely due to a lack of power to detect differences in new task performance: the impact of awareness on the complex task could have been small relative to the overall difficulty and time needed to find PPT-advanced features in the full interface. Beyond using a within-subjects design, one possibility to increase power would be to use a less directed task that would encourage participants to explore more during the simple task, thus magnifying any differences in awareness before starting the complex task. For example, leaving participants to discover commands that will help them replicate a sample final presentation should encourage more exploration of the interface, magnifying potential differences between the conditions.

4.6 Conclusion

In this chapter, we introduced awareness as a new evaluation measure. The combination of awareness and core task performance offers a decomposition of more traditional performance measures. Together, they allow for a more nuanced comparison of different designs. In a controlled laboratory study to evaluate layered interfaces, we demonstrated a measurable trade-off between core task per-

formance and awareness: core task performance in a minimal layered interface approach was better than in the full interface alone, but participants were more aware of advanced features if they had used the full interface from the outset. Previous research on personalization to reduce functionality had largely focused on the benefit of such approaches, including improved initial performance and reduced visual complexity. Our work reveals a more comprehensive understanding of the impact of reducing functionality.

Unfortunately, the two measures of awareness in this study produced inconsistent results. Unlike the recognition test, the new task performance measure provided no support for our hypotheses that the Minimal condition would result in the least awareness, the Control the most, and the Marked condition would be in between. However, since the recognition test scores provided partial support for the hypotheses, we still believed there should be an indirect impact of awareness on performance. Since our inability to detect this difference may have been due to a lack of statistical power for the new task performance measure, we revisited this hypothesis with a more controlled task in the New Task Study (Chapter 6).

Chapter 5

Screen Size Study: Increasing the Benefit of Spatial Adaptation

In addition to introducing and incorporating awareness into our evaluations, we were interested in how specific design characteristics impact the user's experience with spatially adaptive personalized interfaces. In this chapter, we focus on the potential performance benefits that adaptive interfaces provide for small screen devices. Previous research on adaptive interfaces has almost exclusively focused on desktop displays. Here, we present a study that compared the impact of desktop- versus mobile-sized screens on core task performance, awareness, and user satisfaction with adaptive split menus. The primary goal of this study was to show that there is more of a benefit to personalization on the smaller screen than on the desktop-sized screen. In addition, a secondary goal was to extend the results showing a trade-off between core task performance and awareness from the Layered Interface Study (Chapter 4) to a second type of personalized interface, adaptive split menus.

5.1 Introduction and Motivation

With the proliferation of mobile phones and PDAs, small screen devices are now pervasive, but smaller screens can make even basic tasks such as reading and web browsing more difficult [36, 67]. The reduced screen size means that, even with high resolution screens, designers must choose only the most important features to display. Additionally, users tend to use mobile devices in situations where their visual, audio and motor attention is limited in comparison to traditional environments [92], which may make it more difficult for users to navigate a complex interface. To address the limitations of small screen devices, several researchers have proposed that adaptive interfaces, where the system tailors the interface to an individual user's needs, may be beneficial [10, 67].

Despite the potential theoretical benefits, research on adaptation for small screens has focused largely on adaptive web content (e.g., [67, 111]) rather than on adaptive graphical user interface

(GUI) control structures. GUI control structures, such as menus, present unique challenges in comparison to adaptation of content, for example, a higher user expectation for stability [17]. In the context of mobile devices, there has been a small amount of work on adaptive menu structures for phones [13, 3], but evaluations have been informal. The bulk of adaptive GUI research, rather, has been conducted on desktop-sized displays, where evaluations of spatially adaptive techniques have been inconclusive: in some cases, adaptive menus or toolbars have been faster and preferred to their static counterparts [50, 56], whereas other research has shown the opposite [38, 85, 88]. As a result, adaptive GUIs have been conceptually controversial and very few have appeared in commercial applications. If the benefit of adaptivity is more evident for small screens than large screens, adaptivity may be less controversial in this context and should be reconsidered as a viable design alternative.

The main goal of the work reported here was to investigate the impact of a spatially adaptive GUI on small screen displays relative to desktop-sized displays. The results should shed light on the degree to which prior findings directly apply to smaller displays: for instance, an adaptive interface that was less efficient than a static counterpart may no longer be so when the two are used on a smaller screen. Previous work has shown that adaptive accuracy, or the accuracy with which the underlying algorithm can predict the user's needs, can impact performance and satisfaction [50, 124]. We sought to extend this previous work on adaptive accuracy by assessing the potential interaction between adaptive accuracy and screen size. We conducted an experiment with 36 users, comparing adaptive split menus [104] on a desktop screen to a PDA-sized screen. Since adaptive accuracy can affect performance and use of adaptive predictions [50, 124], we included two levels of accuracy (50% and 78%) and a static control condition. Further, we specifically accounted for the predictability and stability within our two accuracy levels, something that has not been done before.

Our study shows that high accuracy adaptive split menus have a larger positive impact on core task performance and user satisfaction in small screens compared to large screens. This suggests that the potential of adaptive interfaces may be best realized in situations where screen real estate is constrained. We had thought this performance and satisfaction differential would be due to reduced navigation (i.e., scrolling) in small screens, but, interestingly, screen size also impacts user behavior: people were more likely to take advantage of the adaptive predictions in the small screen condition. As expected, a low accuracy adaptive interface performs poorly regardless of screen size, which reinforces that research findings on adaptivity must be understood in relation to accuracy levels.

A secondary goal of our work was to extend the results from the Layered Interface Study (Chapter 4) to show that an adaptive personalization approach can also negatively impact awareness because it focuses the user's attention on a small set of frequently used items. Our results show that despite the core task performance benefits of a high accuracy adaptive interface, the personalization does result in reduced awareness, as measured using the recognition test described in Section 4.3.6. The results also suggest that awareness is impacted more negatively in small screens than in large screens. This is an important trade-off that designers will need to consider, especially in light of

findings from 6, showing that lower awareness can negatively impact performance when users are asked to complete new tasks.

This study shows there is a relative benefit of spatially adaptive GUIs for small displays in comparison to large displays, and that this benefit is not purely due to a reduction in the amount of navigation needed to access features, but that screen size also impacts user behaviour. Combined, our findings motivate the need to revisit previous adaptive research in the context of small screen devices, especially for those studies with negative outcomes for adaptive approaches.

5.2 Background

A large body of work exists on usability of non-adaptive small screen interfaces, generally showing that tasks are more difficult on small screens. For example, in comparison to a large screen, reading text requires more navigation [36], and searching the Internet is slower [67]. Related work discussed in Section 2.3.2 is also particularly important for this chapter.

5.2.1 Adaptive Interfaces for Small Screens

Research on adapting GUI control structures has largely been done on desktop displays. One exception is SUPPLE, which automatically adapts interfaces based on device constraints and usage, but evaluations have been small and informal [50]. In other work, Bridle and McCreath compared a static mobile phone menu structure to six approaches that adaptively predicted a single shortcut item [13]. Simulation on logged user data suggested that some of the adaptive approaches would be more efficient than the static one, but no formal user evaluation was reported. Bridle and McCreath stress that stability should be considered in adaptive interface evaluations, which we did for the adaptive menus in our study.

Adaptation of content has been applied more widely to small screens. For example, Smyth and Cotter have used adaptive hypermedia to personalize web portals for mobile devices, showing that personalization can reduce navigation to access content [110]; follow-up large-scale deployment showed that the approach increased customer satisfaction [111]. Adaptation of content, however, may present different challenges than adaptation of control structures [17]. Users may not expect the same degree of spatial stability from content as from control structures, and, compounding this, stability can impact motor memory, one aspect of performance with control structures.

5.2.2 Accuracy of Adaptive Personalization

In the Screen Size Study, we compared static menus to adaptive menus that predict the user's needs with two different levels of accuracy (50% and 78%). Several researchers have previously studied the impact of adaptive accuracy on performance and satisfaction. Tsandilas and Schraefel compared

two approaches for adaptive highlighting of item lists and varied the level of prediction accuracy (100%, 80%, and 60%), finding that the lower accuracy conditions were slower [111]. Results also showed that lower accuracy increased errors for one of the adaptive approaches (Shrink, a fisheye-type distortion), which suggests that the effectiveness of adaptive designs may interact with accuracy.

Gajos, Czerwinski, Tan and Weld compared two adaptive toolbars to a static counterpart within two levels of adaptive accuracy [50]. The two adaptive toolbars were implemented as split interfaces, either moving adaptively suggested items to the adaptive section of the toolbar, or replicating the items there. The accuracy levels were achieved by creating two different tasks for which the algorithms were either 30% or 70% accurate. Results of a controlled experiment showed that the split interface that replicated items was significantly faster than the static toolbar. Both of the adaptive interfaces were faster with the higher accuracy condition, and participants took advantage of the adaptive suggestions more often in that condition.

5.3 Experimental Methodology

To compare the impact of spatially adaptive menus on a small screen versus a desktop-sized display, we conducted a controlled lab study with 36 participants. An obvious drawback of designing for a small screen is that not all items can be shown at once, which results in an added navigation cost for accessing the items that are not immediately available. Our hypothesis was that, by reducing this cost, adaptive interfaces should be relatively more beneficial for a small screen than a large screen. Even so, given that previous results for adaptive GUIs on large screens have been mixed, it was not clear how an adaptive interface for a small screen would compare to a static one.

We chose adaptive split menus for our study because they have been widely studied in the literature, and they appear in commercial applications, such as recency-based font selection menus. We compared two adaptive menus (with 50% and 78% accuracy) and a static menu; the main task was to select a series of menu items. Support for our hypotheses would underscore the need for designers to revisit adaptive approaches in the context of small screens, where they may be more useful.

5.3.1 Conditions

Figure 5.1 shows the layout of the experimental conditions.

Screen Size

To simulate two distinct screen sizes, the window containing the experimental application was either 800x600 pixels (Large screen) or 240x320 pixels (Small screen). For the Large screen, this was big

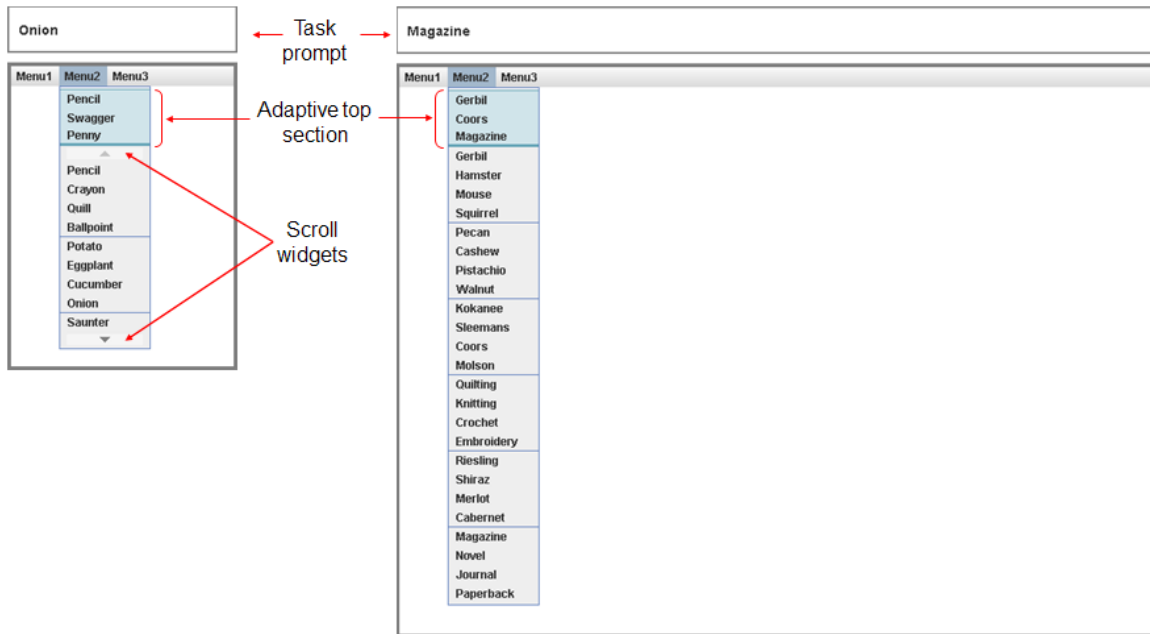


Figure 5.1: Screenshots of Small screen (left) and Large screen (right) experimental setups with adaptive menus open, showing task prompt, adaptive top section, and scroll widgets for the Small screen. The High and Low accuracy adaptive conditions looked identical; the Control condition menu did not have an adaptive top section.

enough to display a full-length menu for our experimental task. The Small screen condition, which was the size of many Pocket PC PDAs, was only big enough to display a subset of menu items at once. To access all items in the Small screen, the user had to hover or click on scroll widgets that appeared at the top and bottom of the menu (similar to menus in Windows Mobile 6). Based on pilot testing with 4 participants, scroll speed was set at 1 item per 75 ms. This was reported to be the best compromise between efficiency and ease of reading item labels; with the faster scrolling speeds pilot participants often overshoot their target and would have to recover by scrolling back. We controlled for input device and display characteristics by using a mouse for both conditions, and simulating the screen sizes on an 18" LCD flat panel monitor with 1280x1024 resolution.

Menu Type

We included a control condition (Control), and High and Low accuracy adaptive conditions. The menus in the Control condition were traditional pull-down menus, while the High and Low adaptive conditions were adaptive split menus. With the split menus, adaptively chosen items were replicated, rather than moved, above the split (as preferred by users in [48]); this necessarily made the split menu slightly longer than the Control menu. The bottom section was identical to the Control menu, while the top section contained three items (as suggested by [31, 104]).

- 1 set top section to the most recently selected item and the two most frequently selected items (as pre-calculated from the selection stream)
- 2 if there is overlap among these three slots or if this is the first selection in the stream (i.e., no recently selected item exists) then the third most frequently selected item is included so that 3 unique items appear in the top
- 3 order top items in the same relative order as they appear in the bottom section of the menu

Figure 5.2: Base adaptive algorithm.

For each condition, the menu bar contained three individual menus, each with 24 items. The 24 items were further separated into semantically related groups of 4 items. (The length and group size were based on averages from four desktop applications, Firefox 2.0, Microsoft Excel 2003, Adobe Reader 7.0, and Eclipse 3.2, including both top-level and cascading submenus.)

Adaptive algorithm detail. To achieve two levels of accuracy, Tsandilas and Schraefel changed the set of adaptive predictions for each trial, either including the item to be selected or not [124]. As acknowledged by the authors, this approach would result in a high level of unpredictability. Gajos et al. took another approach, using two different experimental tasks that resulted in different levels of accuracy for the same interface [50], which makes it difficult to directly compare performance. To address these limitations, we used an identical underlying set of selections for each condition, and determined the adaptive predictions in advance using a two-step process:

1. *Apply base algorithm.* Using a simple base algorithm (shown in Figure 5.2), we pre-calculated the items to appear in the adaptive top. This algorithm incorporated both recently and frequently used items, as suggested by the literature [38, 55] and is commonly used in commercial adaptive user interfaces such as Microsoft Office 2003's adaptive menus. For the randomly generated selection streams in our study (described later), this resulted in 64.2% accuracy on average ($SD = 1.7$).
2. *Adjust accuracy.* To adjust accuracy, we then randomly selected 14% of trials (18 per block, as discussed later) that could be manipulated to increase accuracy (i.e., by swapping the item to be selected into the adaptive top) and 14% that could be manipulated to decrease accuracy (i.e., by swapping the item to be selected out of the adaptive top). This resulted in 50% and 78% accurate adaptive conditions, two somewhat arbitrarily chosen levels of accuracy, as we discuss in Section 5.5. We also enforced several constraints on this manipulation in an effort to maintain spatial stability and predictability (e.g., the most recently selected item always had to appear in the adaptive top).

Spatial stability and predictability of the menus. We chose the above approach because we wanted

	Accuracy		Predictability		Stability	
	<i>M</i> (%)	<i>SD</i> (%)	<i>M</i> (%)	<i>SD</i> (%)	<i>M</i> (%)	<i>SD</i> (%)
Low accuracy	50.0	1.7	94.1	2.3	19.7	2.6
High accuracy	78.5	1.7	94.4	2.0	36.5	3.9

Table 5.1: Average accuracy, predictability, and stability of adaptive conditions. Since task selection streams were randomly generated, values were not identical for each participant ($N = 36$).

the adaptive interfaces to behave as similarly as possible in aspects other than accuracy. We considered: (1) stability, which we defined as the percentage of total trials where no items changed in the adaptive top (similar to [13]), and (2) predictability, which we defined as the percentage of trials where the adaptive top contained the item to be selected, and this could be predicted because that item had been in the adaptive top for the previous trial as well. The accuracy, predictability, and stability of the Low and High accuracy conditions is summarized in Table 5.1. Note that the Low accuracy condition had both lower accuracy and lower stability than the High accuracy condition. While it would have been ideal to achieve the same level of stability for both the High and Low accuracy conditions, this was not possible while using a realistic underlying algorithm, and our compromise at least paired high stability with high accuracy, and vice versa. The relative importance of these factors is discussed in Section 5.5.

5.3.2 Task

The main experimental task was a sequence of menu selections. As shown in the task prompt in Figure 5.1, the system displayed the name of a menu item for each trial but did not specify which menu should be used. Only once a participant had correctly selected the item, the next one would be displayed. To mitigate the impact of any particular set of selections (i.e., item locations), a new set was randomly generated for each participant. However, this underlying set of selections was used for all of an individual participant’s conditions, and different menu masks (or item labels) were applied in each condition to reduce learning effects, similar to previous work [38, 124]. For example, if item 3 on menu 1 was selected first, this was the case for each condition.

The menu masks for each participant were created by randomly assigning 54 semantically related groups of 4 item labels, such that each group appeared once and only once per participant (as in [31]). For example, “diamond, topaz, emerald, sapphire,” represented the precious stones group. All menu item labels were single words, 5-10 letters long. In the Layered Interface Study (Chapter 4) we had reduced the influence of previous experience of menu and toolbar item labels by recruiting only novice PowerPoint users and basing our measures of awareness and core task performance only on interaction with interface items that were specific to PowerPoint. Since the Screen Size Study used a within-subjects design, it was particularly important to be able to create sets of item labels for each condition that would not be unduly or asymmetrically influenced by previous expe-

rience with specific software applications. To address this issue, we chose item labels with which participants would be familiar, but that would not typically be found in a software application.

Previous work has shown both that users only use a small subset of items (for Microsoft Word: 8.7% [79] to 21.5% [87] of items), and that usage can often be modeled by a Zipf distribution [57, 60]. Following the approach of Cockburn, Gutwin and Greenberg [31], we simulated this type of selection pattern: we generated a Zipf distribution (Zipfian $R^2=.99$) across only 8 randomly chosen items out of the 24 items in a menu (with respective frequencies of: 15, 8, 5, 4, 3, 3, 2, 2). The final selection stream was also randomized, for a total of 126 trials per task block (42 trials in each of 3 menus). Each participant completed the same task block twice per condition.

5.3.3 Quantitative and Qualitative Measures

Appendix C contains all questionnaires and the recognition test.

Core Task Performance

Core task performance was measured as time to complete the two task blocks per condition. Error rate was also recorded, although there was an implicit penalty for errors since participants had to correctly complete a trial before advancing.

Awareness

This study was designed to measure awareness using the recognition test introduced in Section 4.3.6. The recognition test used here listed 12 randomly chosen items that were found in the menus for each condition, but were not selected in the tasks. It also included 6 items randomly chosen from a set of distractor items; the full distractor set contained 1 item for each group of 4 items used in the menus, such that the item was related to that group (e.g., distractor for the group “soccer, basketball, baseball, football” was “rugby”). Valid and distractor items were chosen evenly across menus and we calculated corrected recognition rates as discussed in Section 4.3.6. New task performance, our second measure of awareness, was not used in order to keep the study length reasonable.

Subjective Measures

After each menu condition participants were asked to rank the condition along several 7-point Likert scales (with anchors of Disagree, Neutral, and Agree): difficulty, efficiency and satisfaction. Additionally, stability and predictability were also asked for the two adaptive conditions. At the end of the session we asked participants for their overall preference of the three menu conditions.

5.3.4 Design

A 2-factor mixed design was used: screen size (Small or Large) was a between-subjects factor, while menu type (High accuracy, Low accuracy or Control) was within-subjects. Presentation order of menu type was fully counterbalanced.

5.3.5 Participants

Thirty-six participants (19 females) between the ages of 19-49 were randomly assigned to either the Small or Large screen condition and to a presentation order for menu type. Participants were recruited through campus advertising and were screened so that they were not novice computer users (i.e., used a computer for at least 3-5 hours per week). Each participant was reimbursed \$15 to defray any costs associated with their participation.

5.3.6 Apparatus

The experiment used a 2.0 GHz Pentium M laptop with 1.5 GB of RAM, with an 18" LCD monitor at 1280x1024 resolution and Microsoft Windows XP. The application was coded in Java 1.5. Figure 5.1 shows a screenshot of the application: instructions were given one at a time at the top of the screen. The system recorded all timing and error data.

5.3.7 Procedure

The experiment was designed to fit in a 1.5 hour session. Participants were first given a background questionnaire to collect demographic information such as age and computer experience (see Appendix C). Initial pilot participants felt they may have performed differently in the first condition than in subsequent conditions because they did not know to expect a recognition test in the first condition. In response, we included a brief introduction to the format of an awareness-recognition test at the beginning of every session: participants completed a 5-minute paper-based search task on a list of words, followed by an awareness test of words that appeared on the list but were not included in the task.

Following this, the three menu conditions were presented, with 5-minute breaks and paper-based distractor tasks between each. For each condition, the participant completed a short practice block of 15 selections, followed by the same task block repeated twice. To reduce fatigue, 30-second breaks in the middle of each task block and a 1-minute break between blocks were enforced. After the second task block, the awareness recognition test was administered. At the end of all three conditions, a preference questionnaire asked for comparative ratings of the three menu types.

Participants were not told about the different accuracy levels for the conditions. For the first adaptive condition they were simply told that the items in the top section of the menu would change

as they performed the task, and for the second adaptive condition that the behaviour of the top section was slightly different from the previous condition.

5.3.8 Hypotheses

We summarize our main hypotheses:

H1. Core task performance

1. *Higher adaptive accuracy is faster than lower.* The difference between the High and Low accuracy conditions would replicate previous findings [50, 124]. Previous results of comparing adaptive menus to static ones have been conflicting [38, 56, 88, 104], so it was unclear how the static menu would fare.
2. *Small screen is slower than Large screen.* Previous research has shown that tasks such as text reading and content retrieval are slower on small screens (e.g., [36, 67]), so this should be the case for accessing menu items, especially considering the additional scrolling needed.
3. *Effect of adaptive accuracy on core task performance is greater in Small screen than Large screen.* The relative benefit of the adaptive interfaces should be higher for the small screen, largely because they will reduce the amount of scrolling.

H2. Awareness

1. *Higher adaptive accuracy results in lower awareness, and the Control condition has the highest awareness.* The higher the adaptive accuracy, the fewer menu items that users will need to navigate through to complete their task blocks. Thus, higher accuracy should result in reduced awareness.
2. *Small screen results in lower awareness than Large screen.* Since at least half of the menu items are hidden from view at any given time with the Small screen condition, it should result in lower awareness than the Large screen condition.
3. *Effect of adaptive accuracy on awareness is greater in Small screen than in Large screen.* Combining the arguments from H2.1 and H2.2, we would expect the differences in awareness due to accuracy to be even more pronounced in the Small condition.

5.4 Results

A 2x3x2x6 (screen size x menu type x task block x presentation order) repeated measures (RM) ANOVA showed no significant main or interaction effects of presentation order on the main dependent variable of core task performance, and showed a main, learning effect of block. Since both

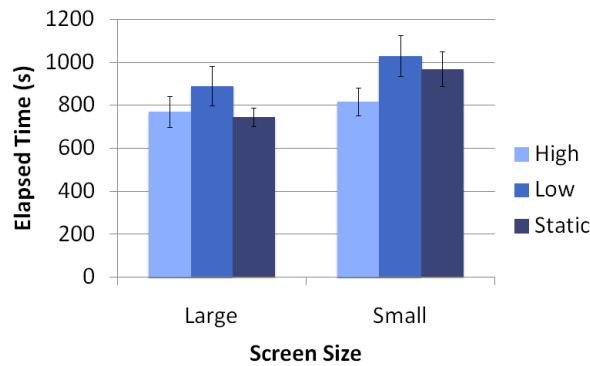


Figure 5.3: Core task performance ($N = 34$). Error bars show 95% confidence intervals.

of these were expected, we simplify our results by examining only effects of screen size and menu type, collapsing across block. All pairwise comparisons were done using t-tests and were protected against Type I error using a Bonferroni adjustment. Where df (degrees of freedom) is not an integer, this is because we have applied a Greenhouse-Geisser adjustment for non-spherical data. We report measures which were significant ($p < .05$) or represent a possible trend ($p < .10$).

Two participants (1 Large screen and 1 Small screen) were removed from the analysis for each having at least one performance measure more than 3 standard deviations away from the mean. Thus, we report on the data of 34 participants.

5.4.1 Core Task Performance

We present the core task performance results first, followed by secondary analysis to understand some of the specific behaviours that may have contributed to the differences in performance (i.e., scrolling and use of adaptive predictions).

Primary Performance Results

On average, participants took 877 seconds to complete both selection blocks in each condition ($SD = 189$). The results are summarized in Figure 5.3. A 2x3 RM ANOVA for performance (screen size x menu type) showed that the combination of menu type and screen had a significant impact on performance (i.e., an interaction effect $F_{2,64} = 9.201, p < .001, \eta^2 = .456$). To understand the reason for this we conducted pairwise comparisons.

High accuracy menus are faster than Low accuracy ones, but outperform the Control condition menus only in Small screens. As predicted by H1.1, the High accuracy condition was faster than the Low accuracy condition in both screen conditions, showing that a higher accuracy interface is more efficient independent of screen size. Support for H1.3 is also shown in the pairwise comparisons by looking at the relative performance of the Control condition to the adaptive menus for each screen

size. The High accuracy condition was no different than the Control condition for the Large screen, whereas it was significantly faster in the Small screen. In contrast, the Low accuracy condition did not perform better than the Control condition with either screen size; in fact, it performed worse than the Control condition in the Large screen. Thus, from a performance standpoint, our results show that there was a benefit to adaptive menus, relative to status quo static menus, only when they have high accuracy and only in Small screens. Low accuracy is at best no worse than traditional menus (for Small screens) and at worst, it degrades performance relative to traditional menus (for Large screens).

Small screen slower than Large screen. As predicted by H1.2, participants were significantly slower using the Small screen, taking 938 seconds on average to complete both task blocks in that condition, compared with 821 seconds in the Large condition (a main effect of screen, $F_{1,32} = 20.923, p < .001, \eta^2 = .395$).

Secondary Analyses: Scrolling and Adaptive Predictions

High accuracy reduces scrolling. One of the expected benefits of the adaptive menus in the Small screen was that they would reduce the amount of scrolling (there was no scrolling in the Large condition). We counted scrolling as the number of items scrolled upward or downward. The mean items scrolled in the High accuracy, Low accuracy, and Control conditions were 1019, 1750, and 1867, respectively. The High accuracy condition indeed resulted in significantly less scrolling than the other two menus, which mirrors the performance results. (A single factor (menu type) RM ANOVA on the Small screen data showed a main effect menu type on scrolling, $F_{2,32} = 31.715, p < .001, \eta^2 = .665$, and $p < .001$ for both the High-Low and High-Control comparisons.)

Small screen increases use of adaptive predictions. Previous work has suggested that lower accuracy adaptive interfaces will result in lower user trust in the adaptive predictions [124], and that users will be less likely to make use of those predictions [50]. To explore this behaviour for the two adaptive menu conditions, we ran a 2x2 (menu type x screen size) RM ANOVA on the percentage of trials where participants did not use the top section of the menu to make a selection that had been correctly predicted by the adaptive menu. We call these non-strategic selections.

Participants in the Large screen condition made significantly more non-strategic selections than participants in the Small screen condition, 22.7% vs. 9.7% (main effect of screen size, $F_{1,32} = 5.706, p = .023, \eta^2 = .151$). This result suggests that participants perceived the adaptive predictions to be more useful in the Small screen condition, which may at least partially explain why the High accuracy menus were faster than the Control condition menus for Small screens but no different for Large screens. Also, as expected, participants made significantly more non-strategic selections in the Low accuracy condition (18.9%) than in the High accuracy condition (11.4%) (a main effect of menu, $F_{1,32} = 7.657, p = .009, \eta^2 = .193$).

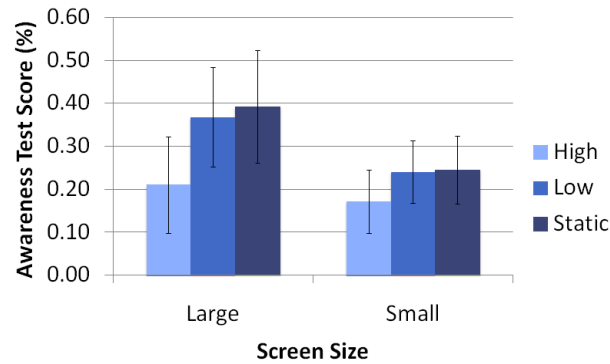


Figure 5.4: Awareness recognition test scores ($N = 34$). Error bars show 95% confidence intervals.

5.4.2 Awareness

After efficiency, we were most interested in how the menu conditions and screen sizes would impact the user's overall awareness of menu items. Figure 5.4 shows the overall corrected awareness test scores. A 2x3 (screen size x menu type) RM ANOVA showed that the menu type did significantly impact users' awareness (main effect of menu type on awareness, $F_{2,64} = 6.547, p = .003, \eta^2 = .170$).

High accuracy results in the lowest awareness. We found partial support for H2.1. As expected, the High accuracy condition had the lowest awareness, with an average score of 19% on the awareness test, in comparison to both the Low accuracy (30%) and Control (31%) conditions (pairwise comparisons were $p = .006$ and $p = .009$, respectively). However, there was no significant difference found between the Low accuracy and Control conditions.

Small screens seem to impact awareness more negatively than Large screens. We found trend level support for H2.2. The Large screen participants scored on average 31% on the awareness test, while the Small participants scored only 22% on average, a difference that was marginally significant (main effect of screen on awareness, $F_{2,32} = 3.392, p = .075, \eta^2 = .096$). However, we did not find any support for H2.3; the different accuracy levels did not have a greater impact on awareness in the Small screen condition relative to the Large condition (there was no significant interaction effect between screen size and menu type, $F_{2,64} = 1.134, p = .328, \eta^2 = .034$).

High accuracy condition fastest for selecting frequent items, but slower than Control condition for infrequent items. As a possible indirect effect of awareness on performance, we wanted to know if participants had more difficulty selecting infrequently accessed items in conditions with lower awareness. To do this, we blocked on *frequency of item*, grouping the 12 items that had been selected only 2 or 3 times per task block separately from the remaining 12 items (i.e., the frequent items) and calculated each participant's average performance for these two groups. This is shown in Figure 5.5. A 2x3x2 (screen size x menu type x frequency block) RM ANOVA on the averages did show that the type of menu differentially impacted both the time it took to select infrequent items

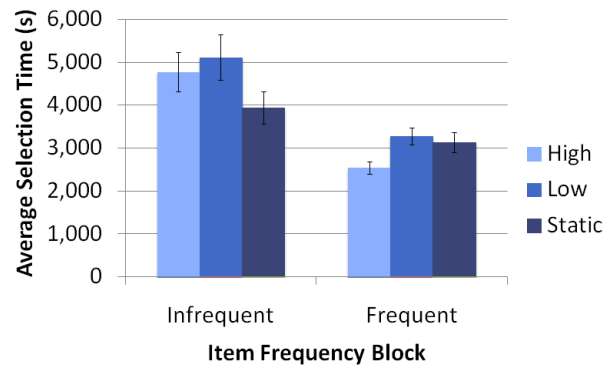


Figure 5.5: Individual selection times of frequently versus infrequently selected items (N =34). Error bars show 95% confidence intervals.

as well as frequent items (a significant interaction effect between menu type and frequency block, $F_{2,64} = 30.365, p < .001, \eta^2 = .487$).

For frequently selected items, the High accuracy condition was faster than the Control and Low accuracy conditions ($p < .001$ for both). However, for the infrequently selected items, the Control condition was faster than both the Low and High accuracy conditions ($p < .001$ for both). This shows that the High accuracy condition made it very efficient to access a small number of features, but the drawback was that it took longer to access the less frequently used features. While this effect may be partly due to the additional visual search time required to process the additional three items in the adaptive conditions, the higher awareness afforded by the Control condition’s menus likely made it easier to learn all the item locations more evenly.

5.4.3 Errors

A 2x3 (screen size x menu type) RM ANOVA showed no significant differences for error rate. Errors were uniformly low in all conditions ($M = 2.2, SD = 1.8$ per condition).

5.4.4 Subjective Measures

High accuracy most satisfying menu in Small screen condition. An internal consistency test showed that our subjective measures of difficulty, efficiency, and satisfaction measured the same internal construct (Cronbach’s alpha = .858). We collapsed these into a single overall satisfaction measure by reversing the scores for the negatively worded question (difficulty) and summing results from all three questions. A 2x3 (screen size x menu type) RM ANOVA showed that overall satisfaction was significantly impacted by a combination of the menu used and the screen size (an interaction effect, $F_{1,643,52.578} = 4.216, p = .027, \eta^2 = .116$). Pairwise comparisons showed that there were no differences in satisfaction for the Large screen. For the Small screen, however, participants

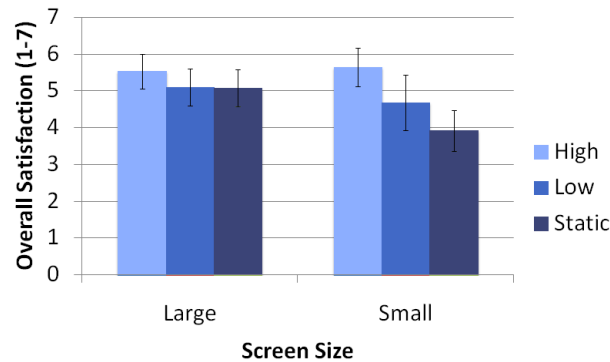


Figure 5.6: Subjective satisfaction ($N = 34$). Error bars show 95% confidence intervals.

were significantly more satisfied with the High accuracy condition than they were with the Low accuracy condition ($p = .008$) and the Control condition ($p < .001$). This pattern reflects the core task performance results and is evident from the data in Figure 5.6.

Participants perceived the High accuracy condition to be more stable and predictable than the Low accuracy condition. Our theoretical calculations for stability and predictability of the menus aligned with participants' perception. A 2x2 (screen size x menu type) RM ANOVA for the High and Low accuracy conditions showed that participants felt that the High accuracy condition was more stable than the Low accuracy condition ($F_{2,32} = 7.493, p = .010, \eta^2 = .190$) and a trend suggested that participants felt that the High accuracy condition was also more predictable than the Low accuracy condition ($F_{2,32} = 3.868, p = .058, \eta^2 = .108$).

High accuracy condition preferred in Small screens, whereas more even split between the High accuracy and Control conditions in Large screens. The majority of participants (12/17) in the Small screen condition chose the High accuracy condition as their preferred menu type. In contrast, preference of Large screen participants was more evenly split between the High accuracy and Control conditions (8 and 6, respectively). Three participants in the Small screen condition chose the Low accuracy condition even though their performance results showed they were faster with the High accuracy condition; when asked afterward to explain their reasoning, they had chosen the Low accuracy condition because they found it more predictable. For the Large screen, 3 participants could not distinguish between the Low and High accuracy conditions; their performance, order of presentation, and non-strategic selections did not provide an obvious explanation for this.

5.4.5 Summary

We summarize our results with respect to our hypotheses.

H1. Core task performance

1. *Higher adaptive accuracy is faster than lower. Supported.* However, performance of the

Control condition relative to the High and Low accuracy conditions depended on screen size.

2. *Small screen is slower than Large screen. Supported.*
3. *Effect of adaptive accuracy on core task performance is greater in Small screen than Large screen. Supported.*

H2. Awareness

1. *Higher adaptive accuracy results in lower awareness, and the Control condition has the highest awareness. Partially supported.* The High accuracy condition resulted in reduced awareness in comparison to the Low accuracy and Control conditions, but there were no differences between the latter two.
2. *Small screen has lower awareness than Large screen. Not supported.* A trend shows this may be supported with more data.
3. *Effect of adaptive accuracy on awareness will be greater in Small screen than Large screen. Not supported.* We found no interaction between screen size and menu condition (accuracy level).

5.5 Discussion

Spatially adaptive interfaces are more beneficial when screen real estate is constrained. Strong evidence shows that the adaptive accuracy conditions fared better in the small screen. The high accuracy adaptive menus were significantly faster (core task performance) and more satisfying than the static menus for the small screen, but these differences disappeared for the large screen. Secondary analyses showed that this was likely due to a combination of the high accuracy condition reducing navigation (i.e., scrolling), and the increased use of adaptive predictions for the small screen. The latter behaviour suggests that users implicitly recognize the added benefit of the adaptive interfaces when screen real estate is constrained. These findings indicate that previous work on adaptive GUIs conducted with desktop applications does not adequately generalize to small screens. Because of the increased potential benefit, researchers and designers should revisit adaptive approaches in the context of reduced screen size.

Adaptive interfaces are low risk for small screens. From a design standpoint, given that it is likely difficult to predict the accuracy of an adaptive interface at design time, our results suggest that there is little performance risk of using adaptive menus in a small screen (for our menu design, as long as the accuracy is at least 50%). For the small screen, the low accuracy adaptive menus were no worse than the static ones, and, if there is potential for them to exhibit higher than 50% accuracy, then from a performance perspective they should be beneficial. For large screens the risk

is much higher: accuracy of 80% provided no performance gain relative to static menus, and 50% degraded performance. As a result, for the adaptive menus to be beneficial on the large screen, the accuracy level would theoretically need to be very high (above 80%). This analysis only considers performance, but subjective measures would need to be considered as well.

Higher accuracy results in reduced awareness. Extending our previous work on awareness (Chapter 4) to an adaptive interface, our results show that the higher accuracy condition resulted in reduced awareness. Perhaps most interesting is that the high accuracy condition had reduced awareness in comparison to the static condition, but it was not faster for the large screen, indicating that it provided no real benefit. This suggests that a static interface may be optimal for the large screen, at least for these two measures. However, for the small screen, the high accuracy condition was significantly faster than the static one, so it may be a better overall choice for the reduced screen size. The differences in awareness also suggest that the designer needs to consider what the goals are for an interface: for example, if the goal is to have the user ultimately become an expert with knowledge of a wide range of features, an interface, such as a static one, that affords higher awareness may be preferred. Alternately, if expertise in a small number of features is sought, then awareness may be less of an issue, and a high accuracy adaptive interface may be preferred.

Sensitivity of awareness measure needs improvement. We had hypothesized that the smaller screen would result in even stronger differences in awareness between the menu conditions. That this did not significantly affect the outcome could be due to a floor effect: the awareness recognition test measure may not have been sensitive enough to detect differences in the small screen condition where awareness scores were all low.

Accuracy, stability, and predictability require more research. In initial pilot testing, 2 out of 4 users commented that our original low accuracy menus were more predictable than the high accuracy menus. Since previous work had not studied the relative impacts of stability, predictability, and accuracy on performance and user satisfaction, we had planned to eliminate a possible confound by creating two accuracy conditions that had similar stability and predictability in the full study. However, given that we required the same task for each condition and had other constraints, such as using a Zipf distribution over items, this was not a straightforward problem. The compromise was to pair higher accuracy with higher stability and lower accuracy with lower stability. As a result, it is unclear whether the poor performance of the low accuracy condition is attributable to accuracy, low stability, or, most likely, to a combination of the two.

Recent work has highlighted the need to report accuracy [50, 124], stability [13, 31], and predictability [51]; our findings stress the importance of all three, along with efficiency. Since we conducted this study, Gajos et al. showed that predictability can have a significant impact on user satisfaction [51], but further work will be needed to understand how much predictability, stability and accuracy separately contribute to performance and satisfaction. For example, a study with fewer task constraints than the one reported here could be designed to include both stability and accuracy

as independent variables. The exact accuracy levels in our study were based on a need to have two reasonable levels that were distinct enough to impact results, but beyond that, they were based on artificial manipulations (similar to [50, 124]). Further work is needed to understand how similar the findings would be for other levels of accuracy.

Adaptive menu models should account for differential usage of adaptive predictions. Cockburn et al. [31] have provided compelling results for modeling adaptive menus. However, their model for adaptive split menus assumes that users will select from the top, adaptive section if the item is there; both our results and those of Gajos et al. [50] show this is not always the case. In addition, Cockburn et al. [31] acknowledge that their model does not incorporate incidental learning (which we measured as awareness). Since an adaptive interface can impact awareness, an obvious extension of the model would be to incorporate awareness.

Generalizability of the results to other GUI control structures. Although further work is needed, the performance and awareness differences between the small and large screens should be equally applicable to other types of GUI control structures, such as toolbars and the MS Office 2007 Ribbon. It is also possible that the particular visual display of features provided in toolbars and the Ribbon will result in similar awareness of the number of features available, but lower awareness of the specific actions that may be carried out by those features, since the images may not as directly convey this information to users as menu labels do.

5.5.1 Limitations of the Experiment

Replication in realistic task context. For a task consisting of only menu item selections, such as the one included in our study, users may be more likely to utilize the adaptive component of the menu because they will value efficiency over other aspects of the interaction. It would be interesting to replicate this work in a more realistic setting where the user's cognitive resources for any given task are divided, and menu selection is but one part of the task. For example, 6/17 participants preferred the static menus in the large screen condition, but in a more realistic setting this may increase. It would also be interesting to study the long-term impact of differences in awareness, such as on an experienced user's ability to complete a new task.

Task appropriate for small screen devices. Further work is needed to understand how our results will apply to tasks specific to mobile computing with small screen devices, and to replicate the work on a mobile device, using pen or stylus input, instead of the simulation we used. Even if mobile application interfaces are simpler than desktop ones, the relative benefit of an adaptive interface may be greater since the user's attention is more fragmented in a mobile context than in a more standard computing context [92].

Differing menu lengths in static and adaptive conditions. The menus in the adaptive conditions were longer than those in the static condition because of the additional three items that were replicated at the top of the menus. It is possible that the different menu lengths may have impacted

performance and awareness. However, we considered this extra length in the adaptive conditions a necessary characteristic of adaptive split menus, so we did not artificially increase the length of the static menus to compensate.

5.6 Conclusion

The Screen Size Study provided empirical evidence to show that high accuracy adaptive split menus may have a larger positive benefit on small screen displays than regular desktop-sized displays. Not only was this shown through direct performance and user satisfaction measures, but we also found that screen size impacts user behaviour: participants were more likely to make use of the adaptive predictions in the small screen condition than the large screen one. We also found that high accuracy adaptive menus negatively impacted the user's overall awareness of features in the interface, as measured using a recognition test of unused features. Finally, our results highlight the importance of considering adaptive performance in relation to accuracy, since the lower and higher accuracy adaptive menus performed differently in relation to their static counterpart when screen size varied.

Overall, these findings stress the need to revisit previous research on spatial adaptive interfaces in the context of small screen devices. Spatial adaptive approaches that may not have been shown to be beneficial on larger screens may be more advantageous in a small screen context. Further work is needed to understand how well our results will generalize in the field, where user tasks are more complex and there are many more demands on the user's attention. Nonetheless, the study presented here provides encouraging evidence that GUI adaptation is a viable design direction for small screen devices.

As a secondary objective, the Screen Size Study extended results from the Layered Interface Study (Chapter 4) by showing a trade-off between core task performance and awareness for a second type of personalized interface. One limitation of the Screen Size Study, however, is that because we wanted to keep the study sessions to a manageable length, we only measured awareness using the recognition test and did not measure new task performance. We address this gap in Chapter 6, which revisits adaptive split menus and measures new task performance.

Chapter 6

New Task Study and the Design Space of Personalized GUIs

The Layered Interface Study (Chapter 4) and the Screen Size Study (Chapter 5) demonstrated a trade-off in personalized interfaces between core task performance and awareness, when measured as the recognition rate of unused features. The study in this chapter revisits the hypothesis, unsupported in the Layered Interface Study, that a reduced level of awareness gained from working in a personalized interface will negatively impact performance when users are asked to complete new tasks. Based on all three studies and a survey of related work, we also discuss the design space of personalized interfaces and present several factors that could affect the trade-off between core task performance and awareness. Finally, we provide a set of design implications that should be considered for personalized interfaces.

6.1 Introduction and Motivation

The Layered Interface Study (Chapter 4) and the Screen Size Study (Chapter 5) demonstrated the trade-off between core task performance and awareness recognition test scores for two different personalization techniques: layered interfaces and adaptive split interfaces, respectively. Since the personalization techniques varied on several design characteristics, including who controlled the adaptation and the frequency with which the interface adapted, those two studies provided a degree of generalizability to the trade-off between core task performance and awareness. However, those studies did not conclusively show whether or not differences in awareness recognition test scores translated into a performance impact when the user was asked to complete new tasks. For the Layered Interface Study this was likely due to a lack of statistical power, while the Screen Size Study was not designed to measure the indirect impact of awareness on performance.

The first goal of this chapter is to build on the results of the previous two studies by revisiting

the hypothesis that personalization will have an impact on performance when users are asked to complete new tasks. To accomplish this goal, we conducted the New Task Study, a controlled lab experiment with 30 participants that used the constrained task and adaptive split menus from the Screen Size Study, but with an experimental design that allowed us to measure an indirect impact of awareness on new task performance. In comparison to our previous attempt to test the hypothesis that awareness impacts new task performance (Chapter 4), using a within-subjects design in the New Task Study meant that participants would have less exposure to each interface. The advantage, however, was that statistical power would be increased.

Awareness and core task performance may be impacted by a variety of personalized interface design factors. A second goal of this chapter is to draw on a survey of related work and our own results from the three controlled laboratory experiments incorporating awareness to outline a design space of graphical user interface (GUI) personalization. We identify four design factors that are particularly important for the interplay between core task performance and awareness: control, granularity, visibility, and frequency of personalization. The design space allows us to identify fruitful areas for future work, and, combined with the study results, we build on it to present several design guidelines for applying personalization approaches.

6.2 Experimental Methodology

The experimental conditions, task and apparatus are largely the same as the Screen Size Study.

6.2.1 Conditions

The experimental conditions each displayed a set of 3 menus, and differed as follows:

1. **High accuracy:** Adaptive split menus that predicted the user's needs with 78% accuracy, on average; that is, 78% of the time the user needed to select an item, that item could be found within the top 3 items in the menu.
2. **Low accuracy:** Adaptive split menus that predicted the user's needs with 50% accuracy, on average.
3. **Control:** Traditional static menus.

The adaptive conditions were the same as those used in the Screen Size Study's large screen condition (Section 5.3.1). Each menu contained 24 items, and the 3 items most likely to be needed by the user were copied to the top of the menu, above the split (see Figure 6.1(a)). We modified the Control condition from the Screen Size Study, however, by adding 3 extra items at the top of the menu, in addition to the 24 regular items (see Figure 6.1(b)). This made the Control condition's menus the same length as the adaptive menus, eliminating menu length as a confound. The extra

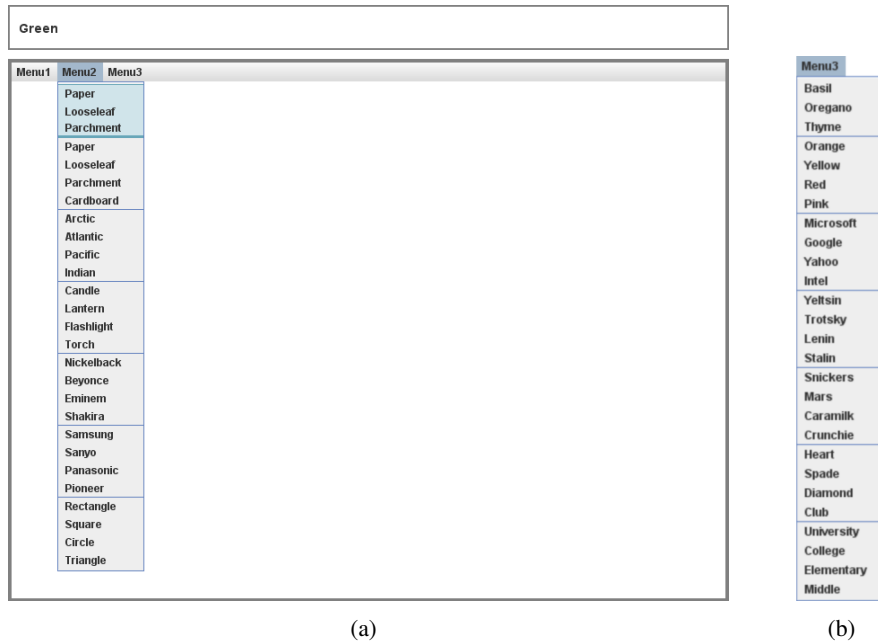


Figure 6.1: Experimental interface: (a) shows the entire setup with an adaptive split menu; (b) shows a static menu from the control condition with 3 extra items at the top (Basil, Oregano, and Thyme).

items were never selected in the experimental tasks and they created a more conservative measure of awareness than using only a 24-item menu for the Control condition: we expected the Control condition to result in the highest awareness, and the additional items should negatively impact awareness-related measures in that condition.

The adaptive algorithm's predictions were based on recently and frequently selected items. To achieve two different levels of adaptive accuracy, we followed the adaptive algorithm and two-step process used in the Screen Size Study to minimize differences between the conditions (see Section 5.3.1 for more detail). First, for each participant we randomly generated a selection sequence (see Section 6.2.2) and applied the adaptive algorithm to predict a set of the 3 items most likely to be needed next by the user; this algorithm resulted in prediction accuracy of 64% on average for all participants. Second, for the Low accuracy condition we randomly adjusted 18 trials so that they were no longer correct, and for the High accuracy condition we randomly adjusted the same number of incorrect predictions to be correct. This resulted in the accuracy conditions listed above.

As with the Layered Interface and Screen Size Studies, we needed users to have no previous experience with the experimental interface in order to accurately measure how the different conditions impacted awareness and performance. Since we also wanted to use a within-subjects design for increased statistical power, we chose to use a custom experimental interface rather than using a real application (unlike in the Layered Interface Study, where we used Microsoft PowerPoint).

This allowed us to create three interface layouts that were similar in every respect other than the personalization mechanism.

6.2.2 Task

The experimental task was a sequence of menu selections. A prompt at the top of the screen displayed the item to be selected by the user, but did not specify in which menu that item would be found (see Figure 6.1(a)). The three menus were located just below the prompt. Once the participant had correctly selected the item, the prompt for the next trial would be shown.

The task was split into two blocks for each condition: a training block and a testing block. The purpose of the training block was to give participants experience with the menus, to develop a base level of awareness that we hypothesized would, in turn, impact performance when selecting new items in the testing block. The training block included selections of only 8 of the 24 items in each menu, whereas the testing block included an additional 4 items in each menu, to simulate an experienced user completing new tasks. The selection sequence for the training block was generated using a Zipf distribution (Zipfian $R^2 = .99$) over 8 randomly chosen items from each menu (i.e., within each menu, the selection frequencies of the 8 items were: 15, 8, 5, 4, 3, 3, 2, 2); this resulted in 126 selections and is the approach taken in the Screen Size Study and used by Cockburn et al. [31]. The testing block was a randomly generated permutation of the exact same set of selections as the training block plus 2 additional selections for each new item, resulting in 150 selections.¹ More detail and rationale for the task and item label selection can be found in Section 5.3.2.

The training and testing selection sequences were randomly generated for each participant to mitigate the effect of a single set of selections. The same underlying sequences for the training block and the testing block were used in each condition for a given participant, but to minimize learning effects the menu items were masked with different labels for each condition. These labels were randomly chosen in groups of 4 semantically related items (e.g., Chardonnay, Shiraz, Merlot, Cabernet) from a larger set of labels such that each label appeared only once for each participant. The labels for the 3 extra items in the Control condition were generated similarly.

In the Screen Size Study, participants completed 252 menu selections before the recognition test. Based on feedback about fatigue from pilot participants and Screen Size Study participants, it was impractical to keep as long a training block as was used in that study when we were additionally asking users to complete a testing block. Instead, participants completed only half as many selections in the training block as compared to the Screen Size Study before taking the recognition test. The implication is that we would not expect as much of an impact on recognition test scores in the New Task Study as in the Screen Size Study.

¹Because of the additional items in the testing block task, the accuracy of the adaptive algorithm necessarily drops slightly.

6.2.3 Design, Participants and Apparatus

The design was within-subjects, with a single factor: menu type (High accuracy, Low accuracy or Control). Order of presentation was fully counterbalanced and participants were randomly assigned to an order. Thirty participants (19 female) aged 19-56 (average 25 years) were recruited through on-campus advertising. Participants were students and community members who were regular computer users. Each participant was reimbursed \$10 per hour to defray any costs associated with their participation.

The experiment used a 2.0 GHz Pentium M laptop with 1.5 GB of RAM, with an 18" LCD monitor at 1280x1024 resolution and Microsoft Windows XP. The application was coded in Java 1.5 and recorded all timing and error data.

6.2.4 Quantitative and Qualitative Measures

Appendix D includes samples of all questionnaires and the recognition test.

Awareness

Our main measure was the impact of awareness on new task performance, defined as the time to select items in the testing block that were not selected in the training block.

We also used a recognition test of unused items to more directly assess awareness (similar to Sections 4.3.6 and 5.3.3). For each participant, this test listed 12 randomly chosen items that were found in the menus for each condition, but were not selected in either the training or testing blocks. It also included 6 items randomly chosen from a set of distractor items; the full distractor set contained 1 item for each group of 4 items used in the menus, such that the item was related to that group (e.g., distractor for the group "soccer, basketball, baseball, football" was "rugby"). Valid and distractor items were chosen evenly across menus.

Core Task Performance

We measured core task performance as the time to select those items in the testing block that had appeared in the training block. Time to select items in the training block was also used as a secondary measure of novice core task performance.

Subjective Measures

Subjective feedback on each of the menu types was collected using 7-point Likert scales anchored with Disagree, Neutral, and Agree. Although there were six questions in total, we were most interested in the first three, which measured awareness-related subjective response: ease of learning the

full set of menu items, ease of selecting infrequent items, and ease of remembering items that were not selected. The remaining Likert scale questions were on efficiency, difficulty, and satisfaction.

6.2.5 Procedure

The study procedure was designed to fit in a single 1.5 to 2 hour session. Participants first completed a background questionnaire to collect demographic information such as age and computer experience (see Appendix D). Then, for each menu condition, participants completed the training block, followed by the paper-based awareness recognition test, then the testing block. Short breaks were given between blocks and between conditions. We collected subjective feedback by questionnaire at the end of each condition and, for comparative comments, at the end of the session.

6.2.6 Hypotheses

Our main hypotheses were:

- H1.** *Impact of awareness on new task performance.* The Control and Low accuracy conditions will be faster than the High accuracy condition (extension of the recognition test results from the Screen Size Study).
- H2.** *Core task performance.* The High accuracy and Control conditions will be faster than the Low accuracy condition, but will be no different from each other (based on the Screen Size Study).
- H3.** *Perception of awareness.* The Control and Low accuracy conditions will be perceived to be easier than High accuracy for the three awareness-related subjective questions (following H1).

6.3 Results

A 3x6 (menu type x presentation order) repeated measures (RM) ANOVA showed no significant main or interaction effects of presentation order on the main dependent variable (new task performance), so we simplify our results by only examining effects of menu type. We ran a one-way RM ANOVA for each of the main dependent measures. All pairwise comparisons were protected against Type I error using a Bonferroni adjustment. One outlier was removed from the analysis for being more than 3 standard deviations away from the mean in one condition for new task performance. We report on results from 29 participants.

6.3.1 New Task Performance

Participants took on average 7.6 minutes to complete the testing block across conditions. As predicted, participants performed poorly with both of the personalized interfaces in comparison to the

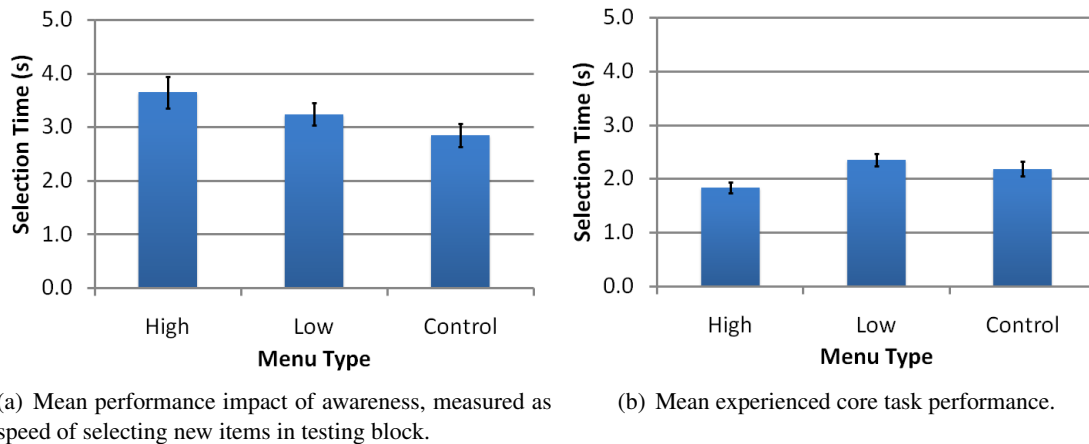


Figure 6.2: New task and core task performance measures ($N = 29$). Error bars show 95% confidence intervals.

Control condition when asked to select new items in the testing block (Figure 6.2(a)): menu type significantly impacted the speed of selecting new items ($F_{2,56} = 21.4, p < .001, \eta^2 = .433$). In the High accuracy condition, participants took on average 3.7 seconds to select a new item, which was significantly longer than the average of 3.2 seconds for the Low accuracy condition ($p = .002$) and the average of 2.9 seconds for the Control condition ($p < .001$). The Control condition was also faster than the Low accuracy condition ($p = .011$). These results reflected our expectations that the Control condition would allow participants to develop a better awareness of the full set of menu items and the location of those items in the interface.

6.3.2 Awareness Recognition Rate

We also analyzed the awareness recognition test scores. On average, test scores for each condition followed the same pattern as the new task performance results; that is, faster performance when selecting new items corresponded to higher scores here. Table 6.1 shows the hit rate, false alarm (distractor) rate and corrected recognition rate from the awareness recognition test. Scores were 20.7%, 24.4% and 27.0% for the High accuracy, Low accuracy, and Control conditions, respectively. Since there were 12 target items on the recognition test, the difference between the means for the Control and High accuracy conditions represents a score difference of approximately 1 item. However, this did not translate to a significant main effect of menu type ($F_{2,56} = .988, p = .379, \eta^2 = .034$) as we found in the Screen Size Study. In retrospect, this is not entirely surprising given that we administered the test after participants had completed only half as many selections as in that study.

	Correct targets (%)	Incorrect distractors (%)	Corrected recognition (%)
High accuracy	28(<i>SD</i> = 19)	9(<i>SD</i> = 11)	21(<i>SD</i> = 18)
Low accuracy	33(<i>SD</i> = 18)	9(<i>SD</i> = 13)	24(<i>SD</i> = 18)
Control	35(<i>SD</i> = 20)	8(<i>SD</i> = 12)	27(<i>SD</i> = 19)

Table 6.1: Average awareness scores as percentage of items answered affirmatively (N = 29).

6.3.3 Core Task Performance

Shown in Figure 6.2(b), there was a main effect of menu type on core task performance of selecting items in the testing block that also appeared in the training block ($F_{2,56} = 58.9, p < .001, \eta^2 = .678$). The High accuracy condition was faster for selecting old items than both the Control condition and the Low accuracy condition ($p < .001$ for both comparisons). Participants were faster in the Control condition than the Low accuracy condition at selecting old items ($p = .002$).

Although it was not one of our main measures, we performed a secondary analysis on core task performance in the training block to assess inexperienced usage. As with the testing block, there was a significant main effect of menu type on speed of selections ($F_{2,56} = 30.7, p < .001, \eta^2 = .523$). The High accuracy condition was faster than both the Control ($p < .001$) and Low accuracy ($p < .001$) conditions. In comparison to the testing block, however, no difference was found between the Control and Low accuracy conditions in the training block.

These results differ from the Screen Size Study, where no difference was found on core task performance between the higher accuracy adaptive menus and the control condition, but both were faster than the lower accuracy adaptive menus. The difference between the High accuracy condition and the Control condition found in the current study is likely due to the extra 3 items included in the Control condition's menus.

6.3.4 Errors

We analyzed testing block error rates separately for newly introduced items and for old items (ones that had appeared in the training block). On average, the error rate was 1.9% across conditions for new items, and there was no significant effect of menu type on error rate ($F_{2,56} = .260, p = .772, \eta^2 = .009$). For the old items, however, there was a significant effect of menu type ($F_{2,56} = 4.23, p = .019, \eta^2 = .131$), but with a Bonferroni adjustment none of the pairwise comparisons were significant. For old items, error rates were 1.4%, 2.1% and 3.0% in the High accuracy, Low accuracy, and Control conditions, respectively.

6.3.5 Subjective Measures

We ran a Friedman test on each of the Likert scale questions and used Wilcoxon signed ranks tests with a Bonferroni adjustment for pairwise comparisons. One participant's questionnaire data was incomplete and is excluded from the analysis.

Subjective responses regarding learning mirrored the new task performance results. Significant differences were found for ease of learning the full set of menu items ($\chi^2_{(2,N=28)} = 9.08, p = .011$) and ease of finding infrequently selected items ($\chi^2_{(2,N=28)} = 12.7, p = .002$). Participants found that the Control condition made it easier to learn the full set of menu items than the Low accuracy condition ($p = .039$), and possibly the High accuracy condition (trend: $p = .093$). It was also easier to select infrequent items with the Control condition than with either the Low ($p = .012$) or High ($p = .021$) accuracy conditions.

The ease of remembering items that were in the menus but that were not selected was also impacted by menu condition ($\chi^2_{(2,N=28)} = 6.50, p = .039$), but no pairwise comparisons were significant. A trend suggested that menu type may impact the efficiency of finding items ($\chi^2_{(2,N=28)} = 5.31, p = .070$). No other significant differences were found.

6.3.6 Summary

We summarize our results according to our hypotheses:

- H1.** *Impact of awareness on new task performance: Supported.* The Control and Low accuracy conditions were faster than the High accuracy condition when selecting new items in the testing block, showing an impact of awareness on new task performance. The Control condition was also faster than the Low accuracy condition.
- H2.** *Core task performance: Partially supported.* The High accuracy and Control conditions were both faster than the Low accuracy condition when selecting old items in the testing block, but, contrary to our hypothesis, the High accuracy condition was also faster than the Control condition.
- H3.** *Perception of awareness: Partially supported.* Participants found the Control condition easiest for selecting infrequent items and a trend suggested this was also the case for learning the full set of menu items. However, the results for ease of remembering unused items were inconclusive, and the Low accuracy condition was not found to be easier than the High accuracy condition for any of the measures.

6.4 Discussion of New Task Study

These findings show that the level of awareness gained from working in a personalized interface impacts the user's performance when completing new tasks; that is, different levels of awareness have the potential to impact future performance. The high accuracy adaptive split menus offered the best core task performance, but also resulted in the worst new task performance; subjective feedback also supported these findings. In comparison to the control condition, the low accuracy adaptive menus resulted in poor performance on both core and new tasks. This supports the Screen Size Study results that show the low accuracy menus do not offer a viable alternative to traditional single-length pull-down menus for desktop-sized screens.

We had expected to find a significant impact of menu type on the recognition test scores in addition to the impact on new task performance. In the Screen Size Study, we found that high accuracy adaptive split menus resulted in lower awareness recognition test scores than both low accuracy adaptive menus and a static control. However, we did not find a significant difference here for the awareness recognition test. Overall, awareness test scores were lower in the New Task Study than in the Screen Size Study (on average, 24% here versus 31% in the large screen condition of the previous study). This is likely due to changes in the study design: we administered the recognition test after the training block, which is half the total time that participants spent in each condition before completing the recognition test in the previous study. The reduced length of exposure to each interface likely explains the lower recognition test scores and the lack of sensitivity of the measure. Descriptively, however, the pattern of mean scores is similar to that found previously, and mirrors the new task performance results. We expect that with a longer training block there would be both higher awareness test scores and statistically significant differences between the three conditions.

Achieving consistent awareness results on both the recognition test and new task performance measures was difficult in the Layered Interface Study and the New Task Study. In the Layered Interface Study, we found statistically significant differences on the recognition test but not on new task performance; the New Task Study yielded the opposite result. This is likely due to the differing primary goals of the two studies and accompanying methodological choices. The Layered Interface Study was designed to provide a more realistic and cognitively demanding experience for participants, so was not optimized to isolate differences in new task performance. It did, however, allow for much longer exposure to the application (42 minutes on average), which may have resulted in more reliable awareness recognition test scores. Because of its more constrained task, the New Task Study was better able to isolate differences in new task performance, but because participants only spent on average 8.5 minutes in each interface condition before completing the awareness recognition test, individual variability obscured possible statistical differences due to menu type. It would be useful to consider in the future whether a standardized memory test could be used as an appropriate covariate to account for some of this variation.

6.5 Personalization Factors Affecting Performance and Awareness

Through a combination of three studies, we have shown that personalized interfaces can positively impact core task performance but negatively impact the user's overall awareness of features in the interface. In turn, these differing levels of awareness can impact new task performance for experienced users, providing an indication of future performance. We explored layered interfaces and adaptive split menus, two personalization approaches that offer contrasting points in the design space of personalized interfaces. However, many other personalization approaches exist, and these approaches could impact the trade-off between core task performance and awareness differently.

Based on a survey of related work, we identify four important design factors for personalized interfaces (see Table 6.2): control, visibility, frequency, and granularity of the adaptation. We discuss these factors in the context of our results and their potential implications for performance and awareness, as well as identifying avenues for further research. Although most previous evaluations of personalized interfaces have not explicitly used the term core task performance, many have measured it. We incorporate results from those studies into our discussion, and, unless otherwise specified, the term *performance* here corresponds to core task performance.

6.5.1 Control of Personalization

Control of the personalization can be adaptable, adaptive, or mixed-initiative, and the choice may impact performance and awareness. We studied an adaptive mechanism (Screen Size Study and New Task Study) and a simulated adaptable mechanism (Layered Interface Study), and found a measurable trade-off between performance and awareness for both.

As seen in previous evaluations of novice performance with personalization mechanisms, the impact of control on performance depends on a number of aspects, for example, for the adaptive approaches, the accuracy of the adaptation (Screen Size Study and [50, 124]), and for adaptable approaches, whether users choose to customize or not. There have been few comparisons of adaptive and adaptable personalization approaches. Comparing adaptive and adaptable split menus, Findlater and McGrenere [38] showed that users were able to personalize their menus effectively, resulting in faster performance than the adaptive counterpart. However, awareness was not distinguished and subsequent research that has improved upon adaptive split interfaces has not compared them to adaptable ones (Screen Size Study and [50]).

The choice of adaptable versus adaptive mechanisms should impact awareness in at least two major respects. First, the cognitive overhead required for the user to adapt an interface, choosing which items to promote or demote, should result in a higher level of awareness of the full set of items than a comparable adaptive approach where this cognition is offloaded to the system. How long this effect lasts beyond the initial adaptation effort, however, would need to be explored. Second, although adaptive approaches to date have been designed with personalization accuracy as the

		Visibility of Change					
		<i>Hidden</i>	<i>Moved</i>	<i>Resized</i>	<i>Replicated</i>	<i>Marked</i>	
Control	<i>Adaptive</i>	<i>Coarse</i>					
		<i>Fine</i>	MS Office 2003 adaptive menus	Original split menus [104], frequency based menus [88], adaptive hierarchical menus [56], adaptive split menus [38] and toolbars [50]	Ability-based interfaces [52], morphing menus [31], walking interfaces [69]	Replicated split interfaces (Chapter 5 and [50, 51])	Colour highlighting [124, 125], ephemeral adaptation (Chapter 7)
	<i>Adaptable</i>	<i>Coarse</i>	Layered interfaces (Chapter 4 and [30, 29, 28, 98, 106], user role-based (Chapter 3 and [54])				Marked layered interface (Chapter 4)
		<i>Fine</i>	Multiple interfaces [85]			Facades [116]	
	<i>Mixed-initiative</i>	<i>Coarse</i>					
		<i>Fine</i>	Incremental interfaces [16], adaptive bar [35], adaptively supported multiple interfaces [20]				

Table 6.2: Design space for personalized GUIs, outlining existing approaches. The adaptive approaches listed here all provide frequent adaptation except for the original split menus [104] and ability based interfaces [52]; the adaptable and mixed-initiative approaches provide persistent adaptation.

main goal, they also have the potential to draw the user’s attention to unused or infrequently used features. Recent recommender system research has begun exploring how recommendations that are not necessarily the most accurate may positively impact the user’s satisfaction [130], a technique that could be explored for adaptive GUIs.

Several characteristics related to control are also important. Accuracy of an adaptive interface can impact core task performance and both the measures of awareness (Screen Size Study and New Task Study). Although the accuracy of user-controlled personalization has not been as widely explored, we expect that similar trends would hold. Other characteristics that should be explored further are the predictability of adaptive personalization, which affects user satisfaction [51], and trust in the personalization. Trust in an adaptive approach’s ability to personalize the interface can

impact user behaviour (Screen Size Study and [124]), but trust may also be a factor with an adaptable approach, since the user needs to be confident in his/her own ability to predict future needs.

6.5.2 Granularity

Fine-grained personalization approaches adjust the interface by manipulating individual features (e.g., [38, 50, 85]), whereas more coarse-grained approaches do so by manipulating larger groups of related features (e.g., [106] and role-based personalization in Chapter 3). Although it does not have to be the case, adaptive techniques have generally been fine-grained, while coarse-grained techniques have been limited to adaptable approaches. Finer-grained approaches should allow for improved performance since they can be more accurately personalized to the user's needs at any given point in time. Coarse-grained approaches, if designed correctly, should be able to contribute to awareness by personalizing the interface to emphasize not only features known to the user, but related features as well.

6.5.3 Visibility of Change

Personalization approaches offer a variety of options for the visual affordance of the adaptation. Some personalization approaches *hide* unnecessary features from view [85, 106], while others *move* [38, 56, 88, 104], *replicate* (Screen Size and New Task Studies and [48, 50]), *mark* (ephemeral adaptation in Chapter 7 and [48, 124]), or *resize* [31, 52] the most salient features to reduce navigation time and/or visual or cognitive complexity. When hiding features, the interface can still provide a degree of visibility, such as the chevron at the bottom of the MS Office 2003 adaptive menus. The early training wheels approach [23, 25] provided *no visual cue* as to which features were blocked; however, because the small amount of more recent work [6] using this technique has yielded negative results, it is not included in Table 6.2.

Hiding, moving, replicating and resizing features are all spatial adaptation techniques. In contrast, the goal of marking techniques is to reduce visual search time by drawing the user's attention to important features. Of all these techniques, completely removing (hiding) the visual affordance associated with blocked features most strongly emphasizes speed of selecting the remaining features. As we saw in the Layered Interface Study, however, there is also an impact on awareness even after transitioning to a more complex interface. The Screen Size Study and the New Task Study demonstrated that replicating features can also impact awareness (at least when personalization accuracy is high).

Marking techniques should not have as much of a negative impact on awareness as spatial adaptation, since all features are as easily visible as in a traditional full interface. In the Layered Interface Study, a trend suggested that the graphical marking technique we used (an 'x') may result in higher awareness than initially hiding features (the minimal layered condition), but we did not find a per-

formance benefit for marking. Other studies have looked at different marking techniques, such as colour highlighting [124] and temporal marking (ephemeral adaptation, Chapter 7), where adaptively predicted items appear briefly before the rest. These may provide more of a performance benefit than the graphical marking technique we used.

With techniques that hide features, the *direction of change* should be considered. An approach like layered interfaces [106] initially provides only a small, core set of features, adding more features as needed. As seen in the Layered Interface Study, this improves core task performance, but also negatively impacts awareness. In contrast, an approach that initially provides the full set of features, then removes unnecessary or unused ones after a period of time, may allow the user to develop greater awareness.

6.5.4 Frequency of Change

Personalization changes may range from as frequent as every user interaction to persistent over much longer-term. In general, adaptive approaches change the interface after every interaction, although this does not have to be the case. For example, the original split menu work only adapted the menus once for a 5 week study [104], and others have proposed personalizing the layout of the interface according to the current document [35]. Adaptable approaches only change as often as the user chooses to make modifications. Realistically, however, it does not make sense for users to adapt the interface after every interaction, so the frequency of change is likely to be less often than with many of the adaptive approaches.

Adapting the interface more frequently should theoretically allow it to more closely match the user's needs at a given point in time, improving performance. Yet, similar to the control of personalization factor, the impact could also be negative if the lack of persistence dominates user performance. Future work should explore how this factor can both positively and negatively impact awareness.

6.6 Design Implications

Based on the results from all three studies incorporating awareness (Layered Interface Study, Screen Size Study and New Task Study) and the discussion in the previous section, we present several guidelines for personalized interfaces.

Look beyond accuracy as the ultimate goal of personalization. Our studies demonstrate the value of including both performance and awareness measures in evaluations of personalized interfaces. The personalized interfaces we studied offered better core task performance, but the trade-off of this improved efficiency for selecting commonly used items is that users are less aware of the full set of features available in the application. Especially for adaptive approaches to personalization,

where much of the focus has been on accuracy, designers need to broaden their focus to consider other aspects of the interaction, including awareness.

Identify the desirable balance between core task performance gains and awareness based on the application context. What is considered to be a desirable balance between core task performance and awareness may depend on different design contexts. High awareness of advanced features will be more important for software applications where users are expected to mature into experts, for example, as with a complex integrated development environment. On the other hand, for applications that are used on a less frequent basis (e.g., many websites) or for those applications that cater to a range of users with varying levels of expertise (e.g., ATMs), the need for efficient performance on core tasks may outweigh the need for awareness.

Match design characteristics to core task performance and awareness goals. We have identified four personalization design factors that are particularly important for performance and awareness (control, visibility, frequency, and granularity). Although more work is needed to map out the impact of all of these factors (and possibly identify further factors), we have provided a first step toward understanding their impact. Designers of personalized interfaces should incorporate design elements that support the particular goals of their system.

Use an appropriate awareness measure in evaluations. We presented two methods for measuring awareness and our experience demonstrates the advantages and disadvantages of each. For a more open-ended task or a field evaluation, the recognition test will be easier to administer because the only requirement is that users need to have had some experience with the interface before completing the recognition test. The performance impact on new task completion is more effortful to apply since it requires the design of an experimental task; however, if the evaluation is in a controlled setting and an appropriate constrained task can be devised, this measure will provide a stronger indication of future performance.

Support exploratory behaviour and make de-emphasized features discoverable. Users often exhibit exploratory behaviour when learning an interface [100], which can be inhibited by personalization. In the Layered Interface Study we saw that users explored more in the control condition than the marked layered condition, even though all features were visible in that condition. A trend also suggested that the control condition facilitated more exploration than the minimal condition. To support exploratory behaviour, especially in cases where features are hidden, users should have an easy means of viewing the full set of features. This should somewhat alleviate concern over hiding features (see Chapter 3 and [87]).

Consider introducing new features to the user. There is the potential for adaptive or mixed-initiative systems to increase the user's awareness of features, by suggesting instances when the user may benefit from unused or underused features (e.g., [16, 79]). Adaptive suggestions have also been used to improve the overall efficiency of user-controlled personalization [20]. Very little work has been done on this type of mixed-initiative interaction, so it is a potentially fruitful area for

further research.

Ultimately, the outcome of an individual design will depend on a number of the above factors and the interaction among them.

6.7 Limitations

The Layered Interface Study in Chapter 4 showed that a minimal layered interface impacts core task performance and awareness recognition test scores in comparison to a static control interface, but no support was found for our hypothesis that the layered interface would also impact performance on new tasks. With the goal of exploring multiple points in the design space, we purposely evaluated different personalization techniques (adaptive split menus) in the Screen Size Study and the New Task Study instead of revisiting this hypothesis for layered interfaces. However, layered interfaces should be revisited. We predict that a task designed to reduce individual variability will yield a statistically significant impact of the minimal layered interface on new task performance.

All three of our studies were conducted in a controlled laboratory setting, where users may value efficiency over longer-term learning. In contrast, in a more realistic setting when cognitive resources are divided among several, complex tasks and GUI feature selection is only part of any given task, users may value a personalization approach that facilitates awareness over one that emphasizes core task performance. A field study would be important for exploring the relationship between performance, awareness, and user satisfaction.

We focused on measuring core task performance and awareness, which we believe are particularly important for interface personalization, but a number of broader challenges need to also be considered when designing a personalization mechanism, as summarized in Chapter 2. For example, adaptive, adaptable, and mixed-initiative mechanisms offer different advantages. Adaptive mechanisms require little or no effort on the part of the user and do not require the user to have specialized knowledge to adapt the interface [43], but have several issues related to lack of user control, unpredictability, transparency, privacy and trust [62]. Adaptable approaches, on the other hand, require effort and motivation on the part of the user to adapt the interface.

6.8 Conclusion

The Layered Interface Study and the Screen Size Study showed that personalization can negatively impact the user's overall awareness of features using both a layered interface and adaptive split menus. In turn, the New Task Study in this chapter showed that personalization also impacts performance on completing new tasks (using adaptive split menus). Although personalization often offers a performance benefit for the user's core tasks, this negative impact on awareness indicates there may be a negative impact on future performance. Based on the study findings and a survey of

related work, we also outlined a design space for personalized interfaces, identifying several factors that are likely to impact performance and awareness, and developed a set of design guidelines.

In the following chapter we turn from the trade-off between core task performance and awareness to introduce ephemeral adaptation, a technique that visually marks adaptively predicted items in the interface. Although we do not measure awareness in our preliminary investigation of ephemeral adaptation, the technique does have the potential to offer a compromise between awareness and core task performance that should be explored in future work.

Chapter 7

Ephemeral Adaptation: Using Gradual Onset to Improve Menu Selection Performance

The Screen Size Study (Chapter 5) showed that the benefit of a spatial personalization approach, such as adaptive split menus, is more likely to outweigh the cost of spatial instability when screen size is constrained. In this chapter, we take a different approach to mitigating the cost of spatial instability: we introduce ephemeral adaptation, a new adaptive graphical user interface (GUI) technique that improves core task performance by reducing visual search time and maintaining spatial stability. Ephemeral adaptive interfaces employ gradual onset to draw the user’s attention to predicted items: adaptively predicted items appear abruptly when the menu is opened, but non-predicted items fade in gradually. To demonstrate the benefit of ephemeral adaptation we conducted two experiments to show: (1) that ephemeral adaptive menus are faster than static menus when accuracy is high, and are not significantly slower when it is low and (2) that ephemeral adaptive menus are also faster than adaptive highlighting. Note that for brevity throughout this chapter, we use the term performance interchangeably with core task performance.

7.1 Introduction and Motivation

Adaptive graphical user interfaces (GUIs) automatically tailor features to better suit the individual user’s needs. To date, these interfaces have tended to rely on one of two forms of adaptation: spatial or graphical. Spatial techniques reorganize items to reduce navigation time and, to a lesser degree, to aid visual search [31, 88, 104]. An adaptive split menu, for example, moves or copies the most frequently and/or recently used items to the top of the menu for easier access [104]. Graphical techniques, on the other hand, reduce visual search time, for example, through changing the background

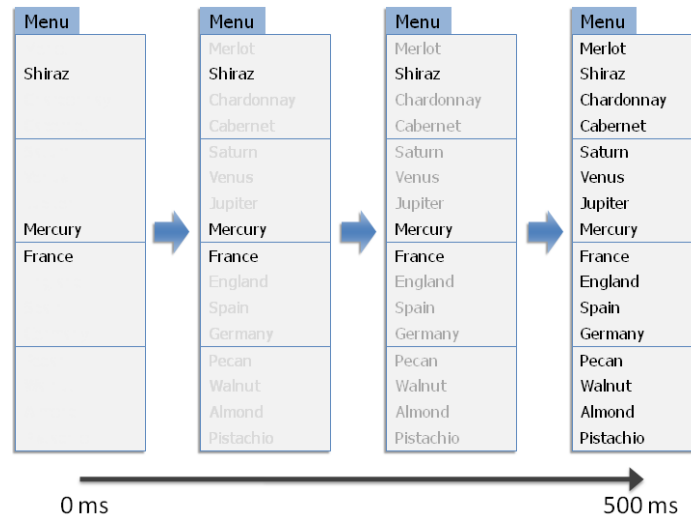


Figure 7.1: Ephemeral adaptation applied to menus: predicted items appear immediately, while remaining items gradually fade in.

colour of predicted items [48, 50, 124]. Some techniques use a combination of both spatial and graphical elements [124, 125].

As an alternative to spatial and graphical adaptation, we propose the use of a temporal dimension and introduce *ephemeral adaptation* as a new adaptive interaction technique that uses this dimension to reduce visual search time. Ephemeral adaptive interfaces use a combination of abrupt and gradual onset to provide initial adaptive support, which then gradually fades away. The goal is to draw the user’s attention to a subset of adaptively predicted items, in turn reducing visual search time. Figure 7.1 applies ephemeral adaptation to a menu: adaptively predicted items appear abruptly when the menu is opened, after which the remaining items gradually fade in.

Ephemeral adaptation maintains spatial stability, thus addressing one of the main drawbacks of spatial adaptation techniques [31]. An adaptive menu that reorganizes features, for example, by promoting the most frequently used ones, offers theoretical performance benefits over a traditional static menu. In practice, however, spatially adaptive interfaces are not often faster than their static counterparts because the user needs to constantly adapt to the altered layout, wiping out any potential gains (Screen Size Study in Chapter 5 and [31, 38, 88]). Successes have tended to occur only when the adaptive approach greatly reduces the number of steps to reach desired functionality, for example, through a hierarchical menu structure [50, 51, 124], or when limited screen real estate necessitates scrolling (Screen Size Study, Chapter 5).

Similarly to ephemeral adaptation, graphical techniques maintain spatial stability and focus on reducing visual search. Several researchers have proposed techniques to highlight predicted items with a different background colour [48, 50, 124] but no performance results comparing colour highlighting to a static control have been reported. Gajos et al. [48] applied colour highlighting to a

graphing calculator and users reported the change to be disorienting [48]; highlighting buttons on the calculator resulted in immediately visible changes, however, in contrast to pull-down menus, where the change is only visible upon opening the menu. Tsandilas and Schraefel [125] also used colour highlighting in bubbling menus, a technique which combines both spatial and graphical elements, but since highlighting was only one aspect of the technique it is not possible to draw conclusions about highlighting alone. While ephemeral and graphical adaptation are similar in that both aim chiefly to reduce visual search time, there is some evidence in the human perception literature that abrupt onset may be a stronger attention cue than colour [120]. This suggests ephemeral adaptation may provide a performance benefit over highlighting.

In this chapter, we introduce ephemeral adaptation, a technique that adapts the interface along a previously little explored temporal dimension. Unlike one previous study [76], which used the abrupt onset of items, we take advantage of findings in the perceptual literature and use gradual onset. To demonstrate the benefit of ephemeral adaptation, we applied the technique to pull-down menus and conducted two controlled lab studies with a total of 48 users. Our results show that when the accuracy with which the adaptive algorithm predicts the user's needs is high (79%), ephemeral adaptation offers performance and user satisfaction benefits over traditional static menus and a performance benefit over an adaptive highlighting technique (based on [50, 124]). Moreover, there is little overall cost to using ephemeral adaptation when adaptive accuracy is low (50%), since ephemeral adaptive menus were not significantly slower than static menus. We also show that adaptive highlighting is not a promising approach for improving performance: although subjective response was positive, highlighting was not found to be faster than static menus even at a high level of adaptive accuracy.

Our results show that ephemeral adaptation is a viable interaction technique to improve visual search time in complex interfaces. While our focus has been on user-adaptive GUIs, the use of the temporal dimension for abrupt and gradual onset should be applicable to a broader range of applications, including guiding attention within visually complex web pages and for information visualization tasks.

7.2 Ephemeral Adaptation

Ephemeral adaptive menus are designed to reduce selection time by guiding the user's attention to adaptively predicted menu items through a combination of abrupt and gradual onset. In contrast to the spatial and graphical techniques described in the previous section, our goal was to design an adaptive mechanism that utilizes a temporal dimension. The adaptation is thus ephemeral and not as intrusive as many adaptive techniques: adaptive support is provided initially but then fades away, returning the interface to normal. This maintains spatial consistency of user interface elements and should reduce visual search time.

7.2.1 Abrupt Onset and Potential Benefit for Adaptive GUIs

Yantis and Jonides [128] demonstrated that an item with an abrupt onset (sudden appearance) is visually processed first among a set of items, even in the absence of an explicit attention set (i.e., the subject has not been told to interpret the stimulus in any specific way). This behaviour results in fast identification of abrupt onset stimuli compared to stimuli without an abrupt onset. In addition, abrupt onset is fairly unique in this regard [68]. Colour can also capture attention, but it is better at doing so when the subject has been instructed to attend to it [45]. Even when such instruction is given, abrupt onset may still be better than colour at drawing attention [120], but this might depend on the particular colour used. Finally, the attention-capturing behavior of abrupt onset can occur below the threshold of subjective awareness [84], suggesting that abrupt onset can be used unobtrusively.

Based on this discussion we predicted that abrupt onset would provide stronger adaptive support than graphical methods for visually distinguishing items, such as background colour highlighting. Even though abrupt onset has the ability to draw attention automatically, research has shown that the response is not involuntary: people can override it if motivated to do so [119, 129]. Thus, if the user knows the abrupt onset stimulus is irrelevant or false, is looking for a different type of stimulus, or knows the location of their target, then an abrupt onset will not distract them. This suggests that using abrupt onset for adaptive predictions should not force the user to give priority to the adaptively predicted items. Thus, when predictive accuracy is low, or the user already knows where the target item is located, the user should not find the ephemeral adaptation approach distracting.

7.2.2 Pilot Testing of Early Designs

We initially tested a design that used two abrupt onsets: adaptively predicted items appeared immediately when the user opened a menu, followed by the abrupt appearance of the non-predicted items after a short onset delay (we piloted onset delays of 25ms, 50ms and 100ms). However, piloting with four participants (using similar methodology to Ephemeral Study 1, described below) was not encouraging. Preliminary analysis of the performance data gave no indication that the adaptive technique reduced selection time, and may even have increased it for some onset delays. Previous work with a menu design that delayed the appearance of some items (though the evaluation did not have it adapt to individual users) has shown a similar negative result [76]. Moreover, 3 out of 4 participants preferred the static control condition to the adaptive conditions.

One possible explanation for this result with the shorter delays (25ms and 50ms) is that users did not notice or did not have enough time to respond to the predicted items before the appearance of the non-predicted items. In contrast, the longer delay (100ms) may have been distracting.

7.2.3 Final Technique

With the final design, non-predicted items faded in gradually over a delay period (as depicted in Figure 7.1). The onset delay for this technique is the elapsed time from opening the menu until all items reach their final foreground colour. The non-predicted items begin as the same colour as the background of the menu (a light grey), and then darken through a series of 10 linear increments until they are the same colour as the predicted items; this gradual appearance is visually smooth for the onset delays we used (250ms to 1000ms). The rationale for this approach is that, unlike abrupt onset, gradual onset does not draw attention [128]. Moreover, because the non-predicted items become legible after only 1 or 2 darkness increments, but the predicted items remain visually prominent until the last 1 or 2 increments, this approach leaves a wider window for the interaction and should allow for more variability among user abilities.

To evaluate this design, we conducted two controlled lab studies. In Ephemeral Study 1, we tuned the onset delay, and as a first step, compared ephemeral adaptation to a static menu. In Ephemeral Study 2, we looked more closely at the performance benefits of ephemeral adaptation by comparing it to a colour highlighting approach. The final design, used in Ephemeral Study 2, has an onset delay of 500ms.

7.3 Ephemeral Study 1: Proof of Concept

The first goal of Ephemeral Study 1 was to determine whether ephemeral adaptation can offer a performance benefit over static menus for a basic selection task. This benefit should be seen when adaptive accuracy is high, and due to the spatial consistency of the menus, there should not be a significant performance cost when accuracy is low. We used pull-down menus since adaptive approaches have been extensively applied to them, facilitating comparison to previous research.

A second, though equally important, goal of this study was to explore different onset delays. Previous research has suggested that 200-300ms should be sufficient to prevent the capture of attention caused by abrupt onsets [12, 122]. However, the task used in that work was quite different from ours (subjects only needed to detect the presence of a stimulus), suggesting a longer delay may be more appropriate for a selection task. Thus, we examined a range of onset delays starting from 250ms, namely 250ms, 500ms, and 1000ms. Early pilot participant feedback for 1000ms was that the delay was too long, so we only looked at the 250ms and 500ms onset delays in Ephemeral Study 1.

7.3.1 Experimental Methodology

Menu Conditions

The three menu types we tested were:

1. **Control:** Traditional static menu.
2. **Short-Onset:** Ephemeral adaptive menu, where non-predicted items gradually appear over a 250ms delay.
3. **Long-Onset:** Ephemeral adaptive menu, where non-predicted items gradually appear over a 500ms delay.

The static control condition (Control) consisted of 3 traditional pull-down menus with 16 items in each menu. Items were separated into groups of 4 semantically related items (e.g., Merlot, Shiraz, Chardonnay, Cabernet). The adaptive conditions were identical to Control, except for the delayed onset of non-predicted items. Menu contents were randomly generated for each participant and condition.

Both adaptive menu conditions used the same adaptive algorithm to predict a set of 3 items that were likely to be selected next by the user; only the onset delay differed. A set size of 3 has been used previously in adaptive split menu research (Chapter 5 and [31, 104]) and is the same number of predictions that are highlighted with similar-length bubbling menus [125].

Adaptive Accuracy Conditions

Adaptive accuracy is the percent of trials where the item the user needs to select is included in the set of predicted items for that trial. We used two levels of adaptive accuracy:

1. **Low:** 50% accuracy, on average.
2. **High:** 79% accuracy, on average.¹

To achieve two different levels of adaptive accuracy, we followed the adaptive algorithm and two-step process used in the Screen Size Study to minimize differences between the conditions (see Section 5.3.1 for more detail). First, for each participant we randomly generated a selection sequence (see the Task section for more detail) and applied an adaptive algorithm to predict a set of 3 probable items at each selection in the sequence; this algorithm calculated predictions based on the items that had been recently and frequently selected and resulted in prediction accuracy of 64.5% on average for all participants. Second, for Low accuracy we randomly adjusted 18 trials so that they were no longer correct, and for High accuracy we randomly adjusted the same number of incorrect predictions to be correct. This resulted in the accuracy conditions listed above.

¹Since the task sequences are randomly generated for each participant as was done in the Screen Size Study and the New Task Study this accuracy level is, by chance, close but not exactly the same as that in the other two studies (78%).

Task

The experimental task was a sequence of menu selections from an experimental system, similar also to that used in the Screen Size Study; more detail and rationale for the task and item label selection can be found in Section 5.3.2. A prompt across the top of the screen displayed the name of the item to be selected and the menu in which it was located. Three menus were positioned just below the prompt. Once the participant had correctly selected the target item, the prompt for the next trial would be displayed.

As described in Section 5.3.2 of the Screen Size Study, the same underlying sequence was used for all conditions and task blocks for a given participant, but the location of the menus was permuted for each condition to reduce learning across conditions. We used a Zipf distribution (Zipfian $R^2 = .99$) over 8 randomly chosen items from each menu (i.e. within a menu the relative selection frequencies of items per block were 15, 8, 5, 4, 3, 3, 2, 2). The final selection sequence consisted of 126 selections per task block and was randomly ordered. Each participant completed two different task blocks per condition.

Quantitative and Qualitative Measures

Speed was measured using the median selection time, calculated as the time from opening the menu to selecting the correct item. The median was used to reduce the influence of outlier trials. We used an implicit error penalty in the speed measures; that is, participants could not advance to the next trial until they correctly completed the current trial. For completeness, we also recorded the error rate. Finally, subjective data was collected using 7-point Likert scales on difficulty, satisfaction, efficiency and frustration. At the end of the study, a questionnaire asked for comparative rankings of the menu conditions. All materials are found in Appendix E.

Apparatus

A 2.0 GHz Pentium M laptop with 1.5 GB of RAM and Microsoft Windows XP was used for the experiment. The system was connected to an 18" LCD monitor with 1280x1024 resolution and the experiment was coded in Java 1.5. The system recorded all timing and error data.

Participants

Twenty-four participants (12 females) were recruited through on-campus advertising. All were regular computer users, were between the ages of 19-45 ($M = 25.5$) and were reimbursed \$10 per hour to defray any costs associated with their participation.

Design

A 2-factor mixed design was used: adaptive accuracy (Low or High) was a between-subjects factor and menu type (Control, Short-Onset or Long-Onset) was a within-subjects factor. Order of presentation was fully counterbalanced and participants were randomly assigned to conditions.

Procedure

The procedure was designed to fit into a single 1-hour session. Participants were first given a background questionnaire to collect demographic information such as age and computer experience (see Appendix E). Then, for each condition participants completed a short 8-trial practice block of selections to familiarize themselves with the behavior of the menus before completing two longer 126-trial task blocks. Short breaks were given in the middle of each block and between blocks. After both task blocks, participants completed a questionnaire with the subjective Likert scale questions for that condition. Once all experimental tasks were complete, a comparative questionnaire was given.

Before each adaptive menu condition, participants were given a brief description of the adaptive behavior: they were told that some of the items would appear sooner than others, and that these were the items the system predicted would be most likely needed by the user. However, participants were not told the level of prediction accuracy.

Hypotheses

The goal of this study was to explore whether differences in onset delay would affect performance, and we did not make any formal hypotheses about the relationship between Long- and Short-Onset. Also, we included Control within each level of accuracy (High vs. Low) along with the Long-Onset and Short-Onset conditions. Since Control is clearly not affected by accuracy level, we did not hypothesize a main effect of accuracy on performance as has been shown in previous work [124, 50].

H1. Speed.

1. *For High accuracy: at least one of Long-Onset or Short-Onset will be faster than Control.* No formal hypotheses for Long- versus Short-Onset.
2. *For Low accuracy: both Long-Onset and Short-Onset will be no worse than Control.* Ephemeral adaptation maintains spatial stability of the menu items, thus we predict that performance should not be significantly hindered when accuracy is low.

H2. User Preference.

1. *For High accuracy: at least one of Long-Onset or Short-Onset will be preferred to Control.* Corresponds to speed hypothesis.

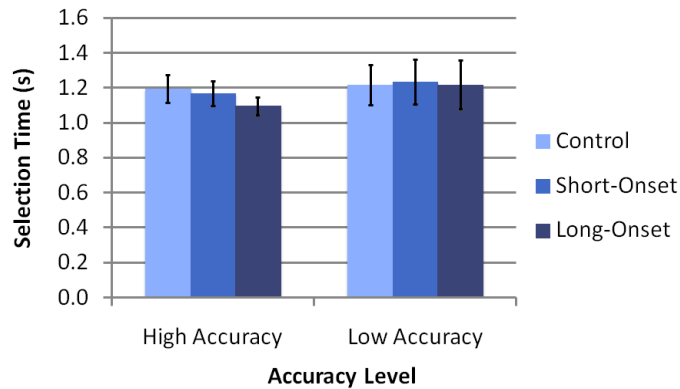


Figure 7.2: Average selection time per trial for Ephemeral Study 1 ($N = 23$). Error bars show 95% confidence intervals.

- For Low accuracy: Control will not be preferred to either Long-Onset or Short-Onset. Corresponds to speed hypothesis.

7.3.2 Results

We ran a $2 \times 3 \times 6$ (accuracy \times menu \times presentation order) repeated measures (RM) ANOVA on the dependent variable of speed. As expected, there were no significant main or interaction effects of order, thus we omit these results from the presentation below. All pairwise comparisons were done using t-tests and were protected against Type I error using a Bonferroni adjustment. Not including break times, the experimental tasks for each condition took on average 10.5 minutes to complete ($SD = 1.5$).

One participant was removed from the analysis. This participant's comments during and at the end of the study indicated that he was confused about the task, particularly in the control condition. Performance-wise, he was more than 2 standard deviations away from the mean on the sum of selection times for all task blocks, and 23% slower than the next slowest participant in his condition (high accuracy). We report on data from 23 participants.

Overall Speed

Selection speeds for each condition are shown in Figure 7.2. Speed was impacted both by menu type (main effect: $F_{2,22} = 3.80$, $p = .038$, $\eta^2 = .257$) and by the combination of menu and adaptive accuracy (interaction effect: $F_{2,22} = 3.73$, $p = .040$, $\eta^2 = .253$). There was no significant main effect of accuracy, as expected.

At High accuracy, Long-Onset was fastest, and at Low accuracy, the adaptive conditions were not slower. Pairwise comparisons on the interaction effect showed that at High accuracy, Long-Onset was faster than both Short-Onset ($p = .018$) and Control ($p = .047$). No significant difference

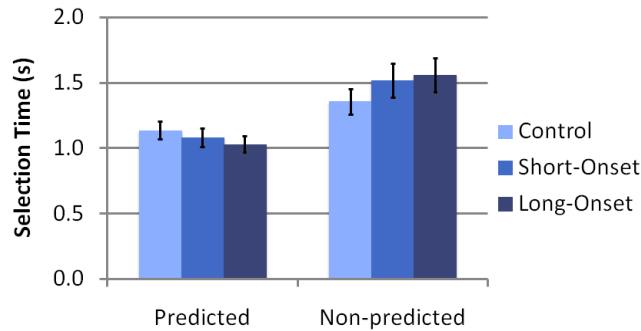


Figure 7.3: Ephemeral Study 1 average selection times on predicted and non-predicted trials, collapsed across accuracy level ($N = 23$). Error bars show 95% confidence intervals.

was found between Short-Onset and Control ($p = .854$). For Low accuracy no differences were found between the three menu types ($p = 1.000$ for all comparisons).

Speed of Selecting Predicted and Non-Predicted Items

While our main measure was overall selection time because it encompasses both the benefit and cost of using ephemeral adaptation, we performed two separate analyses (2x3x6 RM ANOVAs) to better understand this cost/benefit breakdown: (1) speed for trials that were correctly predicted; (2) speed for trials that were not correctly predicted. There were no adaptive predictions in the Control condition, but since each participant's underlying selection stream was the same for every condition, the corresponding Control trials can be compared to the Short-Onset and Long-Onset trials.

Note that we would expect selection times for the non-predicted trials to be longer than for the predicted trials even in the Control condition. This is because the adaptive predictions are based on recently and frequently selected items; thus, non-predicted items are typically those items with which the user is least familiar, and, correspondingly, should be slower selecting. Results are shown in Figure 7.3.

Long-Onset was fastest for predicted trials. For predicted trials there was a significant main effect of menu on speed ($F_{2,22} = 18.5$, $p < .001$, $\eta^2 = .627$). As expected based on the overall results, pairwise comparisons showed that Long-Onset was faster than both Short-Onset ($p = .004$) and Control ($p = .001$). A trend also suggests that Short-Onset was faster than Control ($p = .062$). No significant main or interaction effects of accuracy were found.

Control was fastest for non-predicted trials, suggesting a cost for ephemeral adaptation when items are not correctly predicted. For non-predicted trials there was also a significant main effect of menu on speed ($F_{1,35,14.9} = 10.1$, $p = .001$, $\eta^2 = .479$; Greenhouse-Geisser adjusted). Pairwise comparisons showed that Control was faster than Long-Onset ($p = .008$) and Short-Onset ($p = .047$) for non-predicted trials. No significant difference was found between Long- and Short-Onset ($p = .495$).

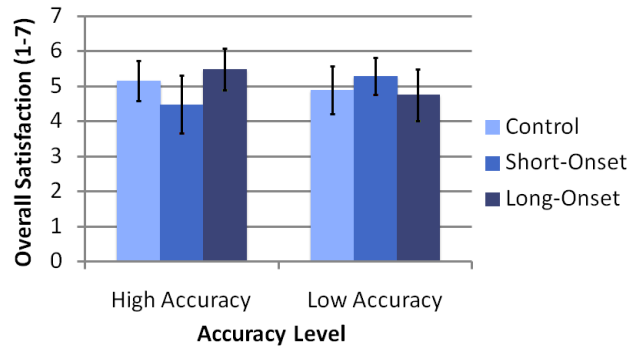


Figure 7.4: Satisfaction ratings for Ephemeral Study 1 ($N = 23$). Higher values indicate higher satisfaction. Error bars show 95% confidence intervals.

Errors

The speed measure included an implicit error penalty but we report on error rates for completeness. Error rates ranged from 1.9%-2.5% of trials on average for participants in the High accuracy conditions and from 0.5%-1.0% on average for the Low accuracy conditions.

Subjective Findings

When we asked participants to rank the menu types based on overall preference, 10 out of 11 High accuracy participants and 9 out of 12 Low accuracy participants chose one of the adaptive conditions. For High accuracy, preference was skewed toward preferring Long-Onset over Short-Onset (7 vs. 3 participants). In contrast, preference was more evenly split in the Low accuracy condition (4 and 5 participants for Long-Onset and Short-Onset, respectively).

A single overall satisfaction measure (shown in Figure 7.4) was constructed by summing the results from the four Likert questions asked for all conditions, reversing the scores where necessary for negatively worded questions. An internal consistency test showed that these questions were likely measuring the same construct (Cronbach's $\alpha = .852$). Friedman tests within accuracy levels showed no significant impact of menu on overall satisfaction. This could be due to low statistical power. Interestingly, in the High accuracy condition the mean rating for Short-Onset was lowest, whereas in the Low accuracy condition it was highest.

7.3.3 Summary and Discussion

We summarize our results according to our hypotheses:

H1. Speed.

1. *For High accuracy: at least one of Long-Onset or Short-Onset will be faster than Control. Supported.* Long-Onset was faster than Control, but no difference was found between Short-Onset and Control.
2. *For Low accuracy: both Long-Onset and Short-Onset will be no worse than Control. Supported.* No differences were found in overall speed.

H2. User Preference.

1. *For High accuracy: at least one of Long-Onset or Short-Onset will be preferred to Control. Not supported.* Although overall satisfaction results were unclear, preference rankings suggest a preference for ephemeral adaptation, which further investigation would need to confirm.
2. *For Low accuracy: Control will not be preferred to either Long-Onset or Short-Onset. Not supported.* Though there was no indication that Control was preferred (supporting our hypothesis), the lack of clear preference results overall suggests this too should be examined further.

Although satisfaction was split between the two ephemeral adaptation approaches, only Long-Onset showed a performance benefit: it was faster than Control at High accuracy and was not slower at Low accuracy. The cost/benefit breakdown of the predicted and non-predicted selections shows that, not unexpectedly, there is a cost to using ephemeral adaptation when items are incorrectly predicted. That this cost did not result in a significant negative impact at Low accuracy suggests it is relatively small in contrast to approaches that do not maintain spatial stability, such as adaptive split menus (Chapter 5 and [38]). However, only for Long-Onset at High accuracy was the benefit for predicted items large enough to provide an overall performance gain.

7.4 Ephemeral Study 2: Ephemeral Adaptation Versus Adaptive Highlighting

Ephemeral Study 2 extends the results from Ephemeral Study 1, comparing the best onset delay condition from that study to an adaptive highlighting approach. We chose highlighting as an appropriate comparison because, like ephemeral adaptation, it maintains the spatial layout of GUI elements and

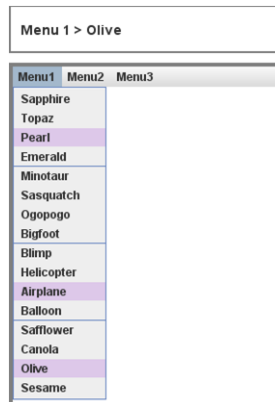


Figure 7.5: Our experimental interface showing the Highlight technique: predicted items have a light purple background.

provides only a visual change. A secondary goal for Ephemeral Study 2 was to evaluate the performance of adaptive highlighting. Although adaptive highlighting had been previously studied in the context of different levels of accuracy, it had not been compared effectively to a control condition.

7.4.1 Experimental Methodology

Ephemeral Study 2 used the same methodology as Ephemeral Study 1 with the exception that Highlight replaced the Short-Onset condition, and to increase the likelihood of finding differences between the menu conditions, we examined only one level of adaptive accuracy. We chose the higher accuracy level from Ephemeral Study 1 because at lower accuracy there was no difference between ephemeral adaptive menus and the static control. Using a high level of accuracy also increased the likelihood of finding a benefit for adaptive highlighting (see Section 7.5 for more detail). The following sections describe the impact of these differences.

Menu Conditions

Ephemeral Study 2 compared three menu types, where both the adaptive menus (Ephemeral and Highlight) used the High accuracy adaptive condition from the previous study:

1. **Control.** The same as Control in Ephemeral Study 1.
2. **Ephemeral.** The 500ms onset delay (Long-Onset) condition from Ephemeral Study 1.
3. **Highlight.** Shown in Figure 7.5, Highlight emphasizes predicted items by changing the background colour to light purple (the same colour as used in [124]). It uses the same adaptive algorithm as Ephemeral.

Participants, Measures and Design

For Ephemeral Study 2, we recruited 24 new participants (10 females). All were regular computer users between the ages 19-33 ($M = 25.3$). A single-factor design was used with menu type (Control, Ephemeral or Highlight) as the within-subjects factor. Order of presentation was fully counterbalanced and participants were randomly assigned to an order. We collected subjective data using the same questionnaires as in Ephemeral Study 1, plus, for the adaptive conditions, 7-point Likert scales on distinctiveness, helpfulness, and distraction. Appendix E contains copies of all questionnaires.

Hypotheses

H1. Speed:

1. *Ephemeral will be faster than Control.* This hypothesis is based on our results from Ephemeral Study 1.
2. *Ephemeral will be faster than Highlight.* Abrupt onset has been shown to be a stronger cue than colour [68, 120]. Although this was shown in a different context, we predict the relationship will also hold here.
3. *Highlight will be faster than Control.* Previous research has not provided definitive results, but has suggested that colour highlighting should offer a performance advantage [48, 124].

H2. Satisfaction/Preference

1. *Ephemeral will be preferred to Control.* Although there were no statistically significant results in Ephemeral Study 1, the descriptive statistics suggested that with a larger sample we may see this result.
2. *Control will be preferred to Highlight.* This hypothesis is based on previous findings [48].
3. *Ephemeral will be preferred to Highlight.* Based on the above two hypotheses, Ephemeral should also be preferred to Highlight.

7.4.2 Results

We ran a 3×6 (menu \times presentation order) RM ANOVA on selection time. As with Ephemeral Study 1, there were no significant main or interaction effects of order. A Bonferroni adjustment was used on all pairwise comparisons. We report on measures that were significant ($p < .05$) or represent a possible trend ($p < .10$). Not including breaks, the experimental tasks for each condition took on average 10.8 minutes to complete ($SD = 1.8$).

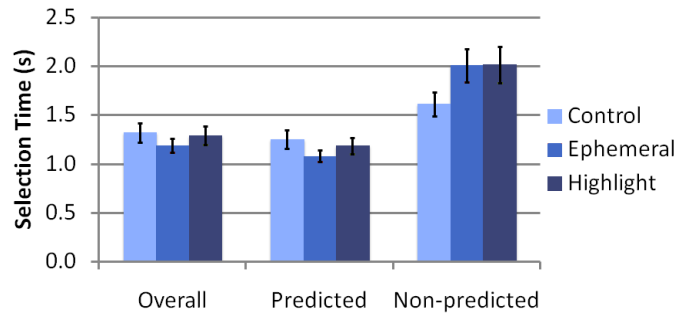


Figure 7.6: Ephemeral Study 2 selection times overall, and for predicted and non-predicted trials. Error bars show 95% confidence intervals ($N = 24$).

Overall Speed

Ephemeral was the fastest menu type overall. There was a significant main effect of menu on speed ($F_{2,36} = 13.4, p < .001, \eta^2 = .427$). Results are shown in Figure 7.6 (Overall). Pairwise comparisons showed that Ephemeral was significantly faster than both Control ($p = .001$) and Highlight ($p = .004$). No significant difference was found between Control and Highlight ($p = .581$).

Speed of Selecting Predicted and Non-Predicted Items

As with Ephemeral Study 1, we performed a secondary analysis, breaking down selections into those that were adaptively predicted and those that were not. Figure 7.6 shows the selection times for predicted and non-predicted trials. There was a significant main effect of menu on speed for both predicted ($F_{2,36} = 21.7, p < .001, \eta^2 = .547$) and non-predicted ($F_{2,36} = 25.7, p < .001, \eta^2 = .588$) trials.

Ephemeral was the fastest menu type for predicted trials. Pairwise comparisons showed that Ephemeral was faster than both Control and Highlight for correctly predicted selections ($p < .001$ for both). Highlight was also faster than Control ($p = .043$).

Control was the fastest for non-predicted trials. Pairwise comparisons showed that Control was faster than both Ephemeral ($p < .001$) and Highlight ($p < .001$) when the predictions were incorrect. No difference was found between Ephemeral and Highlight ($p = 1.000$).

Errors

Error rates were uniformly low, at 2.0%, 2.2% and 2.3% on average for Ephemeral, Highlight and Control, respectively. Errors were indirectly accounted for in our speed measure.

Subjective Findings

Highlight was preferred overall by 12 participants, while 8 preferred Ephemeral and only 4 preferred Static. Most common reasons cited for preferring one of the adaptive conditions included making the task easier or faster.

Ephemeral and Highlight were more satisfying to use than Control. We calculated an overall satisfaction measure similarly to Ephemeral Study 1 (Cronbach's $\alpha = .837$). Ratings were from 1 to 7, where 7 indicated strong positive agreement. A Friedman test showed there was a significant main effect of interface on satisfaction ($\chi^2_{2,N=24} = 12.9, p = .002$). To understand the source of this effect, we performed pairwise comparisons using Wilcoxon Signed Ranks Tests and applied a Bonferroni adjustment. The average satisfaction rating for Control was 4.3 (out of 7), which was less than Ephemeral's average rating of 5.0 ($z = -2.63, p = .027$) and Highlight's average rating of 5.0 ($z = -2.67, p = .024$). No significant difference was found between Ephemeral and Highlight ($p = .871$).

For the two adaptive conditions we also asked participants three additional Likert scale questions. While no statistically significant differences were found (using Wilcoxon Signed Ranks Tests), the ratings were overall positive. Participants felt that both the Ephemeral and Highlight adaptive behaviour helped them distinguish predicted items ($M = 5.5, SD = 1.4$) and helped them find items more quickly ($M = 5.1, SD = 1.4$). Also, participants responded neutrally to whether or not the adaptive behaviour was distracting ($M = 3.6, SD = 1.5$).

7.4.3 Summary

We summarize our results according to our hypotheses.

H1. Speed.

1. *Ephemeral will be faster than Control.* Supported.
2. *Ephemeral will be faster than Highlight.* Supported.
3. *Highlight will be faster than Control.* Not supported. No difference was detected between Highlight and Control for overall performance.

H2. Satisfaction/Preference.

1. *Ephemeral will be preferred to Control.* Supported.
2. *Control will be preferred to Highlight.* Not supported. Contrary to previous results, Highlight was preferred to Control.
3. *Ephemeral will be preferred to Highlight.* Not supported. While more participants preferred Highlight over Ephemeral, no significant differences on overall satisfaction were found between the two conditions.

7.5 Discussion

Our ephemeral adaptation approach, which employs temporal adaptation, shows promise in terms of both performance and user satisfaction. Results show that when predictive accuracy is high (79%), ephemeral adaptation helps users to find adaptively predicted menu items faster than either a static control condition or an adaptive highlighting condition. Another encouraging finding is that, in contrast to our study with adaptive split menus (Chapter 5), the ephemeral conditions did not perform worse than the static control condition when predictive accuracy was low (50%). This suggests that the consistent spatial layout provided by ephemeral adaptation allows its adaptive support to degrade more gracefully with lower accuracy than a split menu. Moreover, users were receptive to the ephemeral adaptive menu and rated it more highly than the static menu. Although further research is required to refine the technique, these combined results suggest that ephemeral adaptation is a viable option for distinguishing adaptive predictions in a visually complex interface.

The adaptive highlighting technique we tested did not show a performance benefit over static menus even though predictive accuracy was high; that is, the small advantage on correctly predicted trials did not translate to a significant overall improvement. However, Highlight was rated by users as comparable to the ephemeral adaptive menu and, in contrast to previous research [48, 50], better than the static menus. One possible explanation is that our implementation was simply more subtle than Gajos et al.'s highlighting techniques (both [48] and [50] used brighter colours). Another possibility is that since Gajos et al.'s evaluations used more complex applications, the highlighting would have competed with other uses of colour in the interface and may have been perceived as more distracting than in our experimental interface. This would suggest that it may be difficult to design visually attractive real world interfaces that use highlighting, and even harder to add colour highlighting to an existing interface.

Any technique that vies for visual attention, including both adaptive highlighting and ephemeral adaptation, will need to compete with other visual elements in the interface. This underscores the need to explore the effectiveness of ephemeral adaptation within the context of a real application where other visual elements, such as animation, may detract from the efficiency and satisfaction benefits seen in our experiment.

In Ephemeral Study 2 we only considered one level of accuracy; thus, we do not have a complete understanding of how colour highlighting compares to ephemeral adaptive menus and static menus. It has, however, already been established for colour highlighting that performance worsens when adaptive accuracy drops [124]. Thus, we would expect that the highlighting technique would at best perform comparably to the static condition when adaptive accuracy is low, and possibly would perform worse. Since the ephemeral adaptive menus were not found to be slower than static menus at low accuracy in Ephemeral Study 1, they likely would not be slower than adaptive highlighting either.

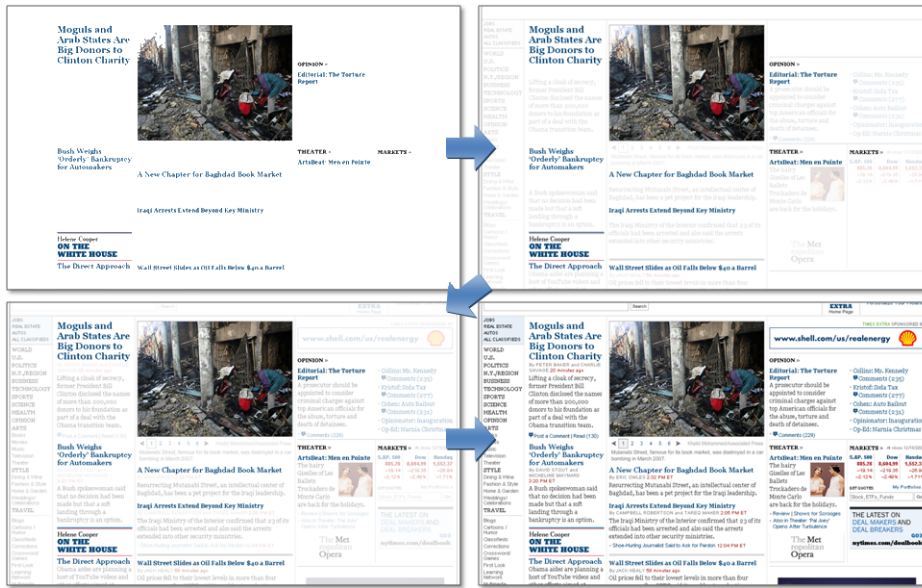


Figure 7.7: Ephemeral adaptation applied to a news website; the headlines appear before the content text and advertisements.

7.6 Applications for Ephemeral Adaptation

Though we focused on pull-down menus, ephemeral adaptation has broader application to a range of interfaces. Most clearly it could be applied to drop-down or tabbed (as in the Microsoft Office Ribbon) toolbars, but it could also be applied to other interface elements that have a point of onset. Conversely, ephemeral adaptation would not be appropriate for visually persistent toolbars.

It could additionally be applicable in contexts that are not necessarily user-adaptive but are visually complex. For example, ephemeral adaptation in a busy webpage like the New York Times homepage could help guide users to content the site designer deems to be important (Figure 7.7). While font size and bolding are commonly used techniques to guide the user's attention and help structure the page, ephemeral adaptation would cause main content to appear abruptly, with the other elements fading in gradually. Of course, a challenge here would be deciding which content should be featured. Similarly, ephemeral adaptation may be useful for guiding the user's attention during information visualization tasks: briefly providing high-level structure before the remaining visualization appears could improve visual processing time.

7.7 Future Work

A number of possibilities for refinement of the ephemeral adaptation technique exist. In Ephemeral Study 2, some participants reported that they would prefer either a longer or shorter onset delay, sug-

gesting that further tuning is needed. In Ephemeral Study 1 and piloting beforehand we determined that the optimal delay is between 250ms and 1000ms, but this range could be narrowed further. Another possible modification is to change the gradual onset function. In all of our ephemeral adaptation conditions, we used a linear darkening algorithm, but other options exist, such as transitioning more slowly through the lighter increments but speeding up for the darker ones, or vice versa.

The optimal onset delay may also depend on the level of adaptive accuracy. We did not see any indication of this in the Ephemeral Study 1 performance results, but we only examined two onset delays thoroughly. Moreover, the satisfaction scores do provide a preliminary indication of an interaction between accuracy and delay length, with the lower accuracy group reporting slightly higher satisfaction for the shorter delay, and the higher accuracy group choosing the longer delay. This makes sense because a longer delay costs more when the prediction is wrong, and this cost could begin to dominate as accuracy falls. Future work could seek to confirm this interaction. If it is true, one possibility to manage the cost/benefit would be to dynamically change the delay onset based on the observed accuracy of the adaptive algorithm.

Future work could also compare ephemeral adaptation to adaptive split menus. However, the similarity of the experimental task used here to the large screen condition and accuracy levels used in the Screen Size Study (Chapter 5) suggests that ephemeral adaptation will be faster than an adaptive split menu (since split menus were not faster than static menus in the Screen Size Study's large screen condition). If anything, since the menus used here were shorter than those used in the Screen Size Study, the reduction in movement time that could be offered by adaptive split menus would be even less.

The studies in this chapter were not designed to measure awareness. Based on the personalized GUI design space discussion in Section 6, we hypothesized that the negative impact of ephemeral adaptation on awareness would be relatively low. Since we already had a lengthy study procedure, we chose not to measure awareness for these initial studies of ephemeral adaptation. Ephemeral adaptation, however, may offer a balance between improved core task performance and reduced awareness in comparison to the adaptive split menus of Chapters 5 and 6, and it will be important to explore this potential in future work.

7.8 Conclusion

We have introduced ephemeral adaptation, a new technique that uses a temporal dimension to reduce visual search time of GUI elements while maintaining spatial consistency. Ephemeral adaptation uses a combination of abrupt and gradual onset to draw the user's attention to the location of adaptively predicted items: when applied to a pull-down menu, predicted items appear abruptly when the menu opens, after which non-predicted items gradually fade in. In contrast to spatial and graphical techniques, which have tended to only find success when adaptation greatly reduces the number of

steps to reach desired functionality, ephemeral adaptation shows promise even for relatively short, single-level pull-down menus.

Through two controlled laboratory studies, we showed that ephemeral adaptation results in both performance and user satisfaction benefits over a static control condition when adaptive accuracy is high, and is no slower when adaptive accuracy is low. We also showed that, at high adaptive accuracy, ephemeral adaptive menus were faster than a colour highlighting technique and both adaptive techniques were preferred to static menus. There was no performance difference between highlighted menus and static menus. The fact that highlighting was liked was surprising considering the lack of performance results and the negative response it has received in previous studies [48, 50].

Combined, our results show that ephemeral adaptation is a promising technique for guiding visual search in complex interfaces. It should be applicable to a broad range of applications because the adaptive support disappears after only a brief delay, allowing for standard interaction with the interface. Ephemeral adaptation may also be useful for visually complex tasks such as scanning a busy web page or navigating information visualizations.

Chapter 8

Conclusion

The goal of this thesis was to improve the design of personalized GUIs by studying user perceptions of personalization, developing an evaluation methodology to more comprehensively capture both the costs and benefits of personalization, and identifying application contexts and design choices that best take advantage of these benefits. In this chapter, we first briefly summarize the steps in this dissertation research before describing the main contributions and outlining promising directions for future work.

To fulfill the thesis goal, we conducted six studies, each examining different personalization approaches or impacts on the user's experience. The Interview Study characterized challenges of coarse-grained personalization and the user's perception of working with such an approach. The findings from this study, in combination with previous work [87], motivated us to run 3 subsequent controlled laboratory studies (Layered Interface, Screen Size and New Task Studies) to empirically measure and provide evidence for a trade-off between the benefit that personalization can provide in terms of core task performance, and the negative impact that it can have on the user's awareness of the full set of features in the interface. Finally, we evaluated the relative benefit of spatially adaptive personalization when screen size is constrained, and we introduced ephemeral adaptation, an adaptive personalization technique that maintains spatial stability of the interface but reduces visual search time.

8.1 Thesis Contributions

8.1.1 Identification of Challenges in Using a Role-Based, Coarse-Grained Personalization Approach

Coarse-grained approaches to personalization have the potential to provide the benefits of adaptable personalization while reducing the burden on the user to personalize at the level of individual features. However, previous research on coarse-grained approaches to personalization, such as layered

interfaces [106] had been limited to relatively simple applications or personalization models. We conducted an interview study (Chapter 3) with 14 users of a commercial integrated development environment that provides role-based personalization. The findings highlight challenges of coarse-grained personalization approaches, including partitioning of features, presentation, and individual differences. These issues should be considered by designers of personalized interfaces, and offer potentially fruitful areas for further research. No evaluations of role-based approaches to reducing GUI complexity had previously appeared in the literature.

8.1.2 Awareness and Core Task Performance

Another contribution of this dissertation is to define, operationalize, and provide empirical evidence for a previously unidentified trade-off between awareness and core task performance in working with personalized GUIs.

Introduction of awareness as a new evaluation metric for personalized designs

To explore the tension between personalizing to improve performance versus the user's ability to learn about new features, we defined feature awareness as a new evaluation measure (Chapter 4). Used alongside core task performance, awareness is operationalized as: (1) the recognition rate of unused features in the interface; and (2) performance on tasks requiring previously unused features. Measuring awareness in conjunction with performance is particularly valuable for personalized interfaces, where the impact on awareness may be greater than in traditional interfaces. Together, awareness and core task performance offer a broader understanding of the impact of working in a personalized interface. Although awareness does not impact performance on routine or known tasks (those supported by the personalization and familiar to the user), it has the potential to impact performance on new tasks. We thus also distinguished core task performance from new task performance.

Trade-off between core task performance and awareness, when measured as the recognition rate of unused features

We conducted two controlled lab studies incorporating awareness and core task performance measures. Each study compared a different type of personalized interface to a control condition and provided empirical evidence of a trade-off between core task performance and awareness, as measured using the recognition test method. The Layered Interface Study (Chapter 4) evaluated a layered interfaces approach to personalization, showing that a minimal layered interface results in better core task performance but lower awareness than a static counterpart. The Screen Size Study (Chapter 5) evaluated an adaptive split menu approach and showed that this trade-off also holds when comparing a high accuracy adaptive split menu to a static menu. These studies allow for a degree of

generalizability and suggest that evaluating awareness will be important for a range of personalized interfaces.

As a secondary contribution, the Layered Interface Study was also the first to compare more than one multi-layer interface. A marked layered interface, where disabled features are visually marked but remain in the interface, provided little benefit over either a minimal layered interface (where disabled features are removed completely) and a static control condition.

Trade-off between core task performance and awareness, when measured as the time to complete new tasks

The New Task Study (Chapter 6) extended results from the Layered Interface and Screen Size Studies, and provides empirical evidence that personalization can impact our second measure of awareness: performance on new tasks. This study differentiated between core task performance and new task performance, showing that participants were fastest at completing new tasks in a static control condition, while a high accuracy adaptive split menu condition provided the best core task performance. A low accuracy adaptive split menu condition provided neither a core task nor a new task performance benefit over the control condition. Although measuring awareness is not a replacement for longer-term field work assessing the impact of a new design on performance, it does provide designers with a low-cost indication of future performance. This should be the case for both our measures of awareness, but new task performance likely provides a stronger indication. Further work will be required to validate that assumption.

A design space for personalized interfaces

Awareness and core task performance may be impacted by a variety of personalized interface design factors. Drawing on a survey of related work and our own results from the three controlled laboratory experiments incorporating awareness (Chapters 4, 5, 6), we outlined a design space of GUI personalization (Chapter 6). We identified four design factors that are particularly important for the interplay between core task performance and awareness: control, granularity, visibility, and frequency of personalization. Based on the design space and our study results, we identified fruitful areas for future work and presented several guidelines to help designers decide whether and how to include GUI personalization in their interfaces.

8.1.3 Cost/Benefit of Adaptive Technique Characteristics

Relative benefit of adaptive GUIs for small displays in comparison to large displays

The Screen Size Study (Chapter 5) provided empirical evidence of the relative benefit of spatially adaptive GUIs for small displays in comparison to large displays. Results showed that adaptive

split menus that predict the user's needs with high accuracy have an even larger positive impact on performance and satisfaction when screen real estate is constrained. A secondary contribution is to show that this benefit is not purely due to a reduction in the amount of navigation needed to access features, but that screen size also impacts user behaviour. Our findings suggest that the benefits of spatial adaptation are more likely to outweigh the costs, such as instability of the interface layout, when it is more difficult to view and navigate to items. This motivates the need to revisit previous adaptive research in the context of small screen devices, especially for those studies with negative outcomes for spatial adaptation.

Ephemeral adaptation, a new adaptive interface technique using abrupt and gradual onset of items

We introduced ephemeral adaptation (Chapter 7), a technique that adapts the interface along a previously little explored temporal dimension: predicted items appear immediately when the menu opens, while the remaining items gradually fade in. Our results provide empirical evidence to show that when the accuracy with which the adaptive algorithm predicts the user's needs is high (79%), ephemeral adaptation offers performance and user satisfaction benefits over traditional static menus and a performance benefit over an adaptive highlighting technique (based on [50, 124]). Moreover, there is little overall cost to using ephemeral adaptation when adaptive accuracy is low (50%), since ephemeral adaptive menus were not significantly slower than static menus. We also showed that adaptive highlighting is not a promising approach for improving performance: although subjective response was positive, highlighting was not found to be faster than static menus even at a high level of adaptive accuracy.

8.2 Directions for Future Research

The discussion and conclusion sections of individual chapters contain ideas for future work. Here, we summarize the directions that are of broadest interest.

8.2.1 Further Work on Awareness

Subjective importance of awareness

We have shown that personalized user interfaces can negatively affect the user's awareness of the full set of features in the interface, but it will be important for future work to study how contextual and user characteristics combine to impact user satisfaction measures related to awareness. What is considered to be a desirable balance between core task performance and awareness may change in different design contexts, as discussed in Section 6.6. High awareness of advanced features may be more important for software applications where users are expected to mature into experts,

for example, as with a complex integrated development environment; in this case, users may be less accepting of personalization mechanisms that negatively impact awareness than in the case of a simpler application that is used less frequently. Another important aspect to explore will be the impact that individual attitudes toward excess features have on a subjective understanding of awareness. McGrenere and Moore [87] point to a spectrum of feature-shy to feature-keen users; feature-keen users, for example, may value awareness more highly.

Generalization to other GUI techniques

It will be important to generalize this research to other GUI interaction techniques, such as the Ribbon in Microsoft Office 2007. The Layered Interface Study included both menus and toolbars, but did not differentiate between the two. In terms of awareness, it is possible that the descriptive nature of text-based control structures (e.g., menus) may result in higher awareness of the actual features in comparison to visual control structures (e.g., toolbars), which may predominantly result only in awareness about the number of available features. The Ribbon, which combines both text and icons, likely provides a different trade-off between core task performance and awareness than is found with either menus or toolbars. It will be important to characterize the differences among these control structures, and to evaluate how they correspond to user satisfaction.

Generalization to other intelligent user interfaces

In the Intelligent User Interface community, some researchers have criticized intelligent systems of “dumbing down” the user when a portion of the user’s cognitive load is offloaded to the system [75]. Intelligent systems can reduce the user’s breadth of experience by reducing opportunities for learning in that domain [66]. Recent research in recommender systems has introduced the notion of topic diversification to improve the user’s experience, in contrast to more narrow definitions of accuracy for recommendation lists [130]. Although differences exist between GUI personalization and content personalization [17], such as for web pages, it will also be interesting to explore parallels in terms of the accuracy of personalization versus the user’s breadth of experience.

Longer-term field study

Although the combination of the Interview Study and the controlled lab studies reported in this dissertation offer differing degrees of ecological validity (e.g., the guided tasks in the Layered Interface Study vs. the simpler menu selection tasks in the other controlled lab studies), a less-controlled field study with real-world tasks will also be needed. Particularly for the more constrained tasks used in Chapters 4 to 7, users may be more likely to value efficiency over longer-term learning. In contrast, when cognitive resources are divided among several, complex tasks, where GUI feature selection is

but one part of any given task, user satisfaction with adaptive interfaces may be affected. A personalization approach that negatively impacts awareness may result in relatively lower user satisfaction because of the potential negative impact on longer-term learning. A longitudinal field study will also be important for assessing how the set of features known to the user is affected over much longer-term by working in a personalized interface.

8.2.2 Further Work on GUI Personalization in General

Exploration of the design space of personalized GUIs

Our design space is not exhaustive, and other possible design factors and extensions will be important to explore in future work. Our studies did not focus on granularity or frequency of change at all. We also largely focused on spatial adaptation approaches, which are likely to have a bigger impact on awareness than graphical or temporal adaptation will have. One exception was the marked interface condition in the Layered Interface Study, which we had hoped would offer a compromise between core task performance and awareness in comparison to the minimal interface condition and the control condition; however, results were inconclusive. Ephemeral adaptation may offer the same kind of compromise between core task performance and awareness that we had hoped to see in the Layered Interface Study. For the initial studies on ephemeral adaptation (Chapter 7) we opted for a simpler study design and procedure over including awareness as a measure, since we hypothesized that the impact on awareness would be quite low. However, future work will need to explore the impact of ephemeral adaptation on awareness.

Extension of ephemeral adaptation

As discussed in Section 7.6, we believe ephemeral adaptation will be useful in a broad range of contexts beyond pull-down menus, and even beyond drop-down or tabbed GUI control panels (toolbars and the Microsoft Office Ribbon). Future work should explore whether the combination of abrupt and gradual onset can be used to reduce search time in other visually complex tasks such as searching a busy webpage or performing information visualization tasks. Further refinement of the onset delay may also increase the value of ephemeral adaptation. The optimal onset delay may be different depending on the task and the accuracy of the adaptive algorithm's predictions. Preliminary satisfaction scores indicated that user satisfaction may be affected by an interaction between accuracy and onset delay; if this is the case then the onset delay could dynamically change based on the observed accuracy.

Investigation of adaptive accuracy levels

The predictive accuracy levels of approximately 50% and 78-79% for the adaptive algorithm we used in the Screen Size Study, the New Task Study, and the Ephemeral Studies are in line with previous research in the area [50, 124, 125], but only represent two points in the spectrum of possible accuracy levels. The research community is far from any definitive understanding of achievable levels of adaptive accuracy for the types of selection tasks used in the Screen Size Study, the New Task Study, and the Ephemeral Studies. A systematic investigation of achievable levels of accuracy and how variable these levels are across applications and across users will be important for future work in the area.

Application to personalized GUI performance models

Cockburn et al. [31] have provided compelling results for modeling adaptive menus and their model could be extended based on our results. Their model for adaptive split menus assumes that users will select from the top, adaptive section if the item is there; both our results and those of Gajos et al. [50] show this is not always the case. In addition, Cockburn et al. acknowledge that their model does not incorporate incidental learning (which we measured as awareness). Since an adaptive interface can impact awareness, an obvious extension of the model would be to incorporate awareness. The model proposed by Hui, Partridge and Boutilier [65] to estimate the cost/benefit of introducing an adaptation into the interface could also be expanded to account for user behaviour with adaptive interfaces where the adaptation is elective (the user can choose to use it or not). Also, both of these models are based on spatial personalization approaches, and would need to be extended to accurately predict performance with graphical and temporal approaches, such as colour highlighting and ephemeral adaptation.

8.3 Concluding Comments

There is a strong tendency to add rather than eliminate features in new versions of software applications, which underscores a major motivation behind GUI personalization. As we move toward web-based applications and cloud computing, the ability for application providers to dynamically change the interface and to easily monitor usage will only increase; this change will facilitate a greater use of personalization and improve its potential for positive impact. We have provided insight into why some GUI personalization approaches fail while others succeed, by showing that personalization approaches that support the user's current task without explicitly introducing new features can negatively impact awareness of unused features, and by studying several contrasting points in the design space of adaptive GUI personalization techniques.

Bibliography

- [1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:734–749, 2005.
- [2] Ignacio Aedo, Susana Montero, and Paloma Díaz. Supporting personalization in a web-based course through the definition of role-based access policies. *Interactive Educational Multimedia*, 4:40–52, 2002.
- [3] Robert St. Amant, Thomas E. Horton, and Frank E. Ritter. Model-based evaluation of expert cell phone menu interaction. *ACM Transactions on Computer-Human Interaction*, 14(1):1, 2007.
- [4] John R. Anderson, Albert T. Corbett, Kenneth R. Koedinger, and Ray Pelletier. Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4(2):167–207, 1995.
- [5] Alan Baddeley. *The Psychology of Memory*. Basic Books, 1976.
- [6] Maria Bannert. The effects of training wheels and self-learning materials in software training. *Journal of Computer Assisted Learning*, 16(4):336–346, 2000.
- [7] Laura Beckwith, Cory Kissinger, Margaret Burnett, Susan Wiedenbeck, Joseph Lawrance, Alan Blackwell, and Curtis Cook. Tinkering and gender in end-user programmers’ debugging. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 231–240, 2006.
- [8] Suresh K. Bhavnani and Bonnie E. John. From sufficient to efficient usage: an analysis of strategic knowledge. In *CHI '97: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 91–98, 1997.
- [9] Suresh K. Bhavnani, Frederick Reif, and Bonnie E. John. Beyond command knowledge: identifying and teaching strategic knowledge for using complex computer applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 229–236, 2001.
- [10] Daniel Billsus, Clifford A. Brunk, Craig Evans, Brian Gladish, and Michael Pazzani. Adaptive interfaces for ubiquitous web access. *Communications of the ACM*, 45(5):34–38, 2002.

- [11] Jan O. Blom and Andrew F. Monk. Theory of personalization of appearance: Why users personalize their PCs and mobile phones. *Human-Computer Interaction*, 18(3):193–228, 2003.
- [12] Bruno G. Breitmeyer and Bela Julesz. The role of on and off transients in determining the psychophysical spatial frequency response. *Vision Research*, 15:411–415, 1974.
- [13] Robert Bridle and Eric McCreath. Inducing shortcuts on a mobile phone interface. In *IUI '06: Proceedings of the 11th International Conference on Intelligent User Interfaces*, pages 327–329, 2006.
- [14] Peter Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110, 2001.
- [15] Peter Brusilovsky, Charalampos Karagiannidis, and Demetrios Sampson. The benefits of layered evaluation of adaptive applications and services. In *Proceedings of UM2001 Workshop on Empirical Evaluation of Adaptive Systems*, pages 1–8, 2001.
- [16] Peter Brusilovsky and Elmar Schwarz. User as student: Towards an adaptive interface for advanced web-based applications. In *Proceedings of the Sixth International Conference on User Modeling*, pages 177–188, 1997.
- [17] Andrea Bunt. *Mixed-Initiative Support for Customizing Graphical User Interfaces*. PhD thesis, University of British Columbia, 2007.
- [18] Andrea Bunt, Cristina Conati, Michael Huggett, and Kasia Muldner. On improving the effectiveness of open learning environments through tailored support for exploration. In *Proceedings of AIED 2001, 10th International Conference on Artificial Intelligence in Education*, pages 365–376, 2001.
- [19] Andrea Bunt, Cristina Conati, and Joanna McGrenere. What role can adaptive support play in an adaptable system? In *Proceedings of the 9th international conference on Intelligent user interface*, pages 117–124, 2004.
- [20] Andrea Bunt, Cristina Conati, and Joanna McGrenere. Supporting interface customization using a mixed-initiative approach. In *Proceedings Intelligent User Interfaces*, pages 92–101, 2007.
- [21] Andrea Bunt, Joanna McGrenere, and Cristina Conati. Understanding the utility of rationale in a mixed-initiative system for gui customization. In *Proceedings of UM 2007: International Conference on User Modeling*, pages 147–156, 2007.
- [22] John M. Carroll and Caroline Carrithers. Blocking learner error states in a training wheels system. *Human Factors*, 26(4):377–389, 1984.
- [23] John M. Carroll and Caroline Carrithers. Training wheels in a user interface. *Communications of the ACM*, 27(8):800–806, 1984.
- [24] John M. Carroll and Mary Beth Rosson. Paradox of the active user. In John M. Carroll, editor, *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, chapter 5, pages 80–111. Bradford Books/MIT Press, 1987.

- [25] Richard Catrambone and John M. Carroll. Learning a word processing system with training wheels and guided exploration. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, pages 169–174, 1987.
- [26] Li Chen and Pearl Pu. Hybrid critiquing-based recommender systems. In *IUI '07: Proceedings of the 12th International Conference on Intelligent User Interfaces*, pages 22–31, 2007.
- [27] Linn Gustavsson Christiernin. Multi-layered design: Theoretical framework and the method in practise. In *Winter Meeting 2005 Proceedings*. Department of Computing Science, Chalmers University of Technology, 2005.
- [28] Linn Gustavsson Christiernin, Rickard Bäckman, Mikael Gidmark, and Ann Persson. iLayer: MLD in an operating system interface. In *AVI '06: Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 87–90, 2006.
- [29] Linn Gustavsson Christiernin, Fredrik Lindahl, and Olof Torgersson. Designing a multi-layered image viewer. In *NordiCHI '04: Proceedings of the 3rd Nordic Conference on Human-Computer Interaction*, pages 181–184, 2004.
- [30] Bryan Clark and Jeanna Matthews. Deciding layers: Adaptive composition of layers in a multi-layer user interface. In *Proceedings HCI International*, 2005.
- [31] Andy Cockburn, Carl Gutwin, and Saul Greenberg. A predictive model of menu performance. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 627–636, 2007.
- [32] Andy Cockburn, Per Ola Kristensson, Jason Alexander, and Shumin Zhai. Hard lessons: effort-inducing interfaces benefit spatial learning. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1571–1580, 2007.
- [33] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum Associates, 2nd edition, 1988.
- [34] Allen Cypher, editor. *Watch what I do: Programming by demonstration*. MIT Press, 1993.
- [35] Matjaz Debevc, Beth Meyer, Dali Donlagic, and Rajko Svecko. Design and evaluation of an adaptive icon toolbar. *User Modeling and User-Adapted Interaction*, 6:1–21, 1996.
- [36] Andrew Dillon, John Richardson, and Cliff McKnight. The effects of display size and text splitting on reading lengthy text from screen. *Behaviour & Information Technology*, 9(3):215–227, 1990.
- [37] The Eclipse Foundation. *Eclipse.org Home*, 2009. Retrieved March 20, 2009 from <http://www.eclipse.org>.
- [38] Leah Findlater and Joanna McGrenere. A comparison of static, adaptive, and adaptable menus. In *CHI '04: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 89–96, 2004.

- [39] Leah Findlater and Joanna McGrenere. Evaluating reduced-functionality interfaces according to feature findability and awareness. In *Proc. IFIP Interact 2007*, pages 592–605, 2007.
- [40] Leah Findlater and Joanna McGrenere. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In *CHI '08: Proceeding of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1247–1256, 2008.
- [41] Leah Findlater, Joanna McGrenere, and David Modjeska. Evaluation of a role-based approach for customizing a complex development environment. In *CHI '08: Proceeding of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1267–1270, 2008.
- [42] Leah Findlater, Karyn Moffatt, Joanna McGrenere, and Jessica Dawson. Ephemeral adaptation: The use of gradual onset to improve menu selection performance. In *Proceeding of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1655–1664, 2009.
- [43] Gerhard Fischer. User modeling in human-computer interaction. *User Modeling and User-Adapted Interaction*, 11(1-2):65–86, 2001.
- [44] Sebastian Fischer and Stephan Schwan. Adaptively shortened pull down menus: Location knowledge and selection efficiency. *Behaviour & Information Technology*, 27(5):439–444, 2008.
- [45] Charles L. Folk, Roger Remington, and James C. Johnston. Involuntary covert orienting is contingent on attentional control settings. *Journal of Experimental Psychology: Human Perception and Performance*, 18(4):1030–1044, 1992.
- [46] Marita Franzke. Turning research into practice: characteristics of display-based interaction. In *CHI '95: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 421–428, 1995.
- [47] Krzysztof Gajos. *Automatically Generating Personalized User Interfaces*. PhD thesis, University of Washington, 2008.
- [48] Krzysztof Gajos, David Christianson, Raphael Hoffmann, , Tal Shaked, Kiera Henning, Jing Jing Long, and Daniel S. Weld. Fast and robust interface generation for ubiquitous applications. In *Proceedings of UbiComp 2005*, pages 37–55, 2005.
- [49] Krzysztof Gajos and Daniel S. Weld. Supple: Automatically generating user interfaces. In *IUI '04: Proceedings of the 9th international conference on Intelligent user interfaces*, pages 93–100, 2004.
- [50] Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. Exploring the design space for adaptive graphical user interfaces. In *AVI '06: Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 201–208, 2006.
- [51] Krzysztof Z. Gajos, Katherine Everitt, Desney S. Tan, Mary Czerwinski, and Daniel S. Weld. Predictability and accuracy in adaptive user interfaces. In *CHI '08: Proceeding of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1271–1274, 2008.

- [52] Krzysztof Z. Gajos, Jacob O. Wobbrock, and Daniel S. Weld. Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. In *CHI '08: Proceeding of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1257–1266, 2008.
- [53] Dina Goren-Bar, Ilenia Graziola, Fabio Pianesi, and Massimo Zancanaro. The influence of personality factors on visitor attitudes towards adaptivity dimensions for mobile museum guides. *User Modeling and User Adapted Interaction*, 16(1):31–62, 2005.
- [54] Saul Greenberg. Personalizable groupware: Accomodating individual roles and group differences. In *Proceedings of the European Conference of Computer Supported Cooperative Work (ECSCW '91)*, pages 24–27, 1991.
- [55] Saul Greenberg. *The computer user as toolsmith: The use, reuse, and organization of computer-based tools*. Cambridge University Press, 1993.
- [56] Saul Greenberg and Ian Witten. Adaptive personalized interfaces: A question of viability. *Behaviour and Information Technology*, 4(1):31–45, 1985.
- [57] Saul Greenberg and Ian H. Witten. Directing the user interface: How poeple use command-based computer systems. In *Proceedings of the IFAC 3rd Man Machine Systems Conference*, pages 349–356, 1988.
- [58] Tovi Grossman, Pierre Dragicevic, and Ravin Balakrishnan. Strategies for accelerating on-line learning of hotkeys. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1591–1600, 2007.
- [59] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. A survey of software learnability: Metrics, methodologies and guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 649–658, 2009.
- [60] Stephen José Hanson, Robert E. Kraut, and James M. Farber. Interface design and multivariate analysis of unix command use. *ACM Transactions on Information Systems*, 2(1):42–57, 1984.
- [61] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.
- [62] Kristina Höök. Steps to take before intelligent user interfaces become real. *Journal of Interacting with Computers*, 12(4):409–426, 2000.
- [63] Eric Horvitz. Principles of mixed-initiative user interfaces. In *CHI '99: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 159–166, 1999.
- [64] James I. Hsia, Elspeth Simpson, Daniel Smith, and Robert Cartwright. Taming Java for the classroom. *SIGCSE Bulletin*, 37(1):327–331, 2005.
- [65] Bowen Hui, Grant Partridge, and Craig Boutilier. A probabilistic mental model for estimating disruption. In *IUI '09: Proceedings of the 13th International Conference on Intelligent User Interfaces*, pages 287–296, 2008.

- [66] Anthony Jameson. Adaptive interfaces and agents. In Julie A. Jacko and Andrew Sears, editors, *The human-computer interaction handbook, 2nd Edition*, pages 433–458. Lawrence Erlbaum Associates, 2008.
- [67] Matt Jones, George Buchanan, and Harold Thimbleby. Improving web search on small screen devices. *Interacting with Computers*, 15(4):479–795, 2003.
- [68] John Jonides and Steven Yantis. Uniqueness of abrupt onset in capturing attention. *Perception and Psychophysics*, 43(4):346–354, 1988.
- [69] Shaun K. Kane and Jacob O. Wobbrock Ian E. Smith. Getting off the treadmill: evaluating walking user interfaces for mobile devices in public spaces. In *MobileHCI '08: Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 109–118, 2008.
- [70] Judy Kay. Learner control. *User Modeling and User-Adapted Interaction*, 11:111–127, 2001.
- [71] Caitlin Kelleher and Randy Pausch. Stencils-based tutorials: design and evaluation. In *CHI '05: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 541–550, 2005.
- [72] Mik Kersten and Gail C. Murphy. Mylar: A degree-of-interest model for IDEs. In *AOSD '05: Proceedings of the 4th International Conference on Aspect-Oriented Software Development*, pages 159–168, 2005.
- [73] Andy J. Ko, Brad A. Myers, and Htet Htet Aung. Six learning barriers in end-user programming systems. In *2004 IEEE Symposium on Visual Languages and Human Centric Computing*, pages 199–206, 2004.
- [74] Thomas K. Landauer. *Chapter 9: Behavioral research methods in human-computer interaction*, pages 203–227. Elsevier Science B.V., 2nd edition, 1997.
- [75] Jaron Lanier. Agents of alienation. *interactions*, 2(3):66–72, 1995.
- [76] Dong-Seok Lee and Wan Chul Yoon. Quantitative results assessing design issues of selection-supportive menus. *International Journal of Industrial Ergonomics*, 33:41–52, 2004.
- [77] Wai On Lee and Philip J Barnard. Precipitating change in system usage by function revelation and problem reformulation. In *Proceedings of HCI '93*, pages 35–47, 1993.
- [78] D. Leutner. Double-fading support - A training approach to complex software systems. *Journal of Computer Assisted Learning*, 16(1):347–357, 2000.
- [79] Frank Linton, Deborah Joy, Hans-Peter Schaefer, and Andrew Charron. Owl: A recommender system for organization-wide learning. *Educational Technology & Society*, 3(1):62–76, 2000.

- [80] Wendy E. Mackay. Patterns of sharing customizable software. In *Proceedings of the 1990 ACM Conference on Computer-Supported Cooperative Work*, pages 209–221, 1990.
- [81] Wendy E. Mackay. Triggers and barriers to customizing software. In *CHI '91: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 153–160, 1991.
- [82] Allan MacLean, Kathleen Carter, Lennart Lövstrand, and Thomas Moran. User-tailorable systems: Pressing the issues with buttons. In *CHI '90: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 175–182, 1990.
- [83] Mark T. Maybury and Wolfgang Wahlster, editors. *Readings in intelligent user interfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [84] Peter A. McCormick. Orienting attention without awareness. *Journal of Experimental Psychology: Human Perception and Performance*, 23(1):168–180, 1997.
- [85] Joanna McGrenere, Ron Baecker, and Kellogg Booth. An evaluation of a multiple interface design solution for bloated software. In *CHI '02: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 163–170, 2002.
- [86] Joanna McGrenere, Ron M. Baecker, and Kellogg S. Booth. A field evaluation of an adaptable two-interface design for feature-rich software. *ACM Transactions on Computer-Human Interaction*, 14(1):3, 2007.
- [87] Joanna McGrenere and Gale Moore. Are we all in the same “bloat”? In *Proceedings of Graphics Interface*, pages 187–196, 2000.
- [88] Jeffrey Mitchell and Ben Shneiderman. Dynamic versus static menus: An exploratory comparison. *SIGCHI Bulletin*, 20(4):33–37, 1989.
- [89] Bonnie A. Nardi. *A Small Matter of Programming*. MIT Press, 1993.
- [90] Reinhard Oppermann. Adaptively supported adaptability. *International Journal of Human-Computer Studies*, 40(3):455–472, March 1994.
- [91] Antti Oulasvirta and Jan Blom. Motivations in personalisation behaviour. *Interacting with Computers*, 20:1–16, 2007.
- [92] Antti Oulasvirta, Sakari Tamminen, Virpi Roto, and Jaana Kuorelahti. Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile hci. In *CHI '05: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 919–928, 2005.
- [93] Stanley R. Page, Todd J. Johnsgard, Uhl Albert, and C. Dennis Allen. User customization of a word processor. In *CHI '96: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 340–346, 1996.
- [94] Jungchul Park, Sung H. Han, Yong S. Park, and Youngseok Cho. Usability of adaptable and adaptive menus. In *Proceedings HCII 2007*, pages 405–511, 2007.

- [95] Stephanie Parkin. *Rapid Java and J2EE Development with IBM WebSphere Studio and IBM Rational Developer*. IBM, 2004. Retrieved February 3, 2009 from <http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/wp-radrwd-medres.pdf>.
- [96] Tim F. Paymans, Jasper Lindenberg, and Mark Neerincx. Usability trade-offs for adaptive user interfaces: ease of use and learnability. In *IUI '04: Proceedings of the 9th International Conference on Intelligent User Interfaces*, pages 301–303, 2004.
- [97] James Pitkow, Hinrich Schütze, Todd Cass, Rob Cooley, Don Turnbull, Andy Edmonds, Eytan Adar, and Thomas Breuel. Personalized search. *Communications of the ACM*, 45(9):50–55, 2002.
- [98] C. Plaisant, H. Kang, and B. Shneiderman. Helping users get started with visual interfaces: Multi-layered interfaces, integrated initial guidance and video demonstrations. In *Proceedings HCI International 2003: Volume 4 Universal Access in HCI*, pages 790–794. Lawrence Erlbaum Associates, 2003.
- [99] Chris Quintana, Joseph Krajcik, and Elliot Soloway. A case study to distill structural scaffolding guidelines for scaffolded software environments. In *CHI '02: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 81–88, 2002.
- [100] John Rieman. A field study of exploratory learning strategies. *ACM Transactions on Computer-Human Interaction*, 3(3):189–218, 1996.
- [101] Diego Rivera. The effect of content customization on learnability and perceived workload. In *CHI '05: CHI '05 Extended Abstracts on Human Factors in Computing Systems*, pages 1749–1752, 2005.
- [102] Mary Beth Rosson. Patterns of experience in text editing. In *CHI '83: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 171–175, 1983.
- [103] Mary Beth Rosson, John M. Carroll, and Rachel K. E. Bellamy. Smalltalk scaffolding: a case study of minimalist instruction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 423–430, 1990.
- [104] Andrew Sears and Ben Shneiderman. Split menus: Effectively using selection frequency to organize menus. *ACM TOCHI*, 1(1):27–51, 1994.
- [105] Ben Shneiderman. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Intelligent User Interfaces*, pages 33–39, 1997.
- [106] Ben Shneiderman. Promoting universal usability with multi-layer interface design. In *Proceedings of the 2003 Conference on Universal Usability*, pages 1–8, 2003.
- [107] Ben Shneiderman and Patti Maes. Direct manipulation vs. interface agents: Excerpts from debates at IUI 97 and CHI 97. *Interactions*, 4(6):42–61, 1997.
- [108] Ben Shneiderman and Catherine Plaisant. The future of graphic user interfaces: Personal role managers. In *HCI '94: Proceedings of the Conference on People and computers IX*, pages 3–8, 1994.

- [109] Randall B. Smith, Ranald Hixon, and Bernard Horan. Supporting flexible roles in a shared space. In *CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 197–206, 1998.
- [110] Barry Smyth and Paul Cotter. Personalized adaptive navigation for mobile portals. In *Proc. European Conference on Artificial Intelligence (ECAI 2002)*, pages 608–612, 2002.
- [111] Barry Smyth, Paul Cotter, and Stephen Oman. Enabling intelligent content discovery on the mobile internet. In *Proc. AAI*, pages 1744–1751, 2007.
- [112] Rodolfo Soto. Learning and performing by exploration: label quality measured by latent semantic analysis. In *CHI '99: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 418–425, 1999.
- [113] D. Stamoulis, P. Kanellis, and D. Martakos. Tailorable information systems: Resolving the deadlock of changing user requirements. *Journal of Applied System Studies*, 2(2), 2001.
- [114] Margaret-Anne Storey, Daniela Damian, Jeff Michaud, Del Myers, Marcellus Mindel, Daniel German, Mary Sanseverino, and Elizabeth Hargreaves. Improving the usability of Eclipse for novice programmers. In *Eclipse '03: Proceedings of the 2003 OOPSLA Workshop on Eclipse Technology eXchange*, pages 35–39, 2003.
- [115] Anselm C. Strauss and Juliet Corbin. *Basics of qualitative research: Grounded theory procedures and techniques*. Sage, 1990.
- [116] Wolfgang Stuerzlinger, Olivier Chapuis, Dusty Phillips, and Nicolas Roussel. User interface façades: Towards fully adaptable user interfaces. In *UIST '06: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, pages 309–318, 2006.
- [117] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 449–456, 2005.
- [118] Loren Terveen, Jessica McMackin, Brian Amento, and Will Hill. Specifying preferences based on user history. In *CHI '02: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 315–322, 2002.
- [119] Jan Theeuwes. Exogenous and endogenous control of attention: The effect of visual onsets and offsets. *Perception and Psychophysics*, 49(1):83–90, 1991.
- [120] Jan Theeuwes. Stimulus-driven capture and attentional set: Selective search for color and visual abrupt onsets. *Journal of Experimental Psychology: Human Perception and Performance*, 20(4):799–806, 1994.
- [121] Christopher G. Thomas and Mette Krogsøter. An adaptive environment for the user interface of excel. In *IUI '93: Proceedings of the 1st International Conference on Intelligent User Interfaces*, pages 123–130, 1993.

- [122] David J. Tolhurst. Reaction times in the detection of gratings by human observers: A probabilistic mechanism. *Vision Research*, 15:1143–1149, 1974.
- [123] Robert Trevelyan and Dermot P. Browne. A self-regulating adaptive system. In *CHI '87: Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, pages 103–107, 1987.
- [124] Theophanis Tsandilas and m. c. schraefel. An empirical assessment of adaptation techniques. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 2009–2012, 2005.
- [125] Theophanis Tsandilas and m. c. schraefel. Bubbling menus: A selective mechanism for accessing hierarchical drop-down menus. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1195–1204, 2007.
- [126] Jim Warren. Cost/benefit based adaptive dialog: Case study using empirical medical practice norms and intelligent split menus. *Australian Computer Science Communications*, 23(5):100–107, 2001.
- [127] Stephan Weibelzahl and Gerhard Weber. Advantages, opportunities, and limits of empirical evaluations: Evaluating adaptive systems. *Künstliche Intelligenz*, 3:17–20, 2002.
- [128] Steven Yantis and John Jonides. Abrupt visual onset and selective attention: Evidence from visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 10(5):601–621, 1984.
- [129] Steven Yantis and John Jonides. Abrupt visual onsets and selective attention: Voluntary versus automatic allocation. *Journal of Experimental Psychology: Human Perception and Performance*, 16(1):121–134, 1990.
- [130] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *WWW '05: Proceedings of the 14th International Conference on World Wide Web*, pages 22–32, 2005.

Appendix A

Interview Study Materials

Glossary:

- RAD: IBM Rational Application Developer
- WSAD: IBM WebSphere Application Developer (RAD's predecessor)

Contains:

- Background questionnaire
- Sample interview questions

Background Questionnaire

1. Briefly, what are your main job responsibilities, especially related to development?

2. How long have you been developing software in general? (in years) _____
3. How long have you been doing Java development? (in years) _____
4. How long have you been working with WSAD or RAD? (in years) _____
5. How long have you been working with RAD 6? (in months) _____
6. How many hours per week do you spend developing with WSAD/RAD? _____
7. What types of applications do you develop with WSAD/RAD?

8. How many projects do you participate in at one time? _____
9. Would you consider yourself a novice, intermediate or expert user of WSAD or RAD? _____
10. How many additional plug-ins do you have installed? (Go to Help -> Software Updates -> Manage Configuration) _____
11. If you open the preferences dialog and click on Capabilities, you'll find a two-level hierarchy of capabilities that can be enabled and disabled. In the top level of this hierarchy there are 12 categories, for example, Java Development.
How many of those have a black checkmark? _____
How many of those have a grey checkmark? _____
12. How many perspectives are listed if you go to Window -> Open Perspective -> Other..., then click on Show All? _____
 - If you have saved customized perspectives, how many on this list are perspectives you have created? _____
13. In the perspective you use the most often, how many views do you usually have visible at once (i.e., only views which are fully visible, not just accessible by a tab)? _____

Sample Interview Questions

General Customization Practice and User Model

- 1.1. Which additional plug-ins have you installed for RAD, if any? (Are these optional components [_____] or 3rd party plugins [_____])
- 1.2. When installing these plug-ins, do you use update manager or do you install them yourself?
- 1.3. When or how often do you install new plug-ins? Do you do this when you first install RAD or do you do it as you go along and realize you want some new functionality? [_____]
- 1.4. Do you ever disable plug-ins? What has prompted you to remove plug-ins in the past?
- 1.5. I'm trying to get an overall picture of how you customize RAD for your own personal work practice and preference, things like changing font size, code formatting rules, including new views. Can you give me a brief, general overview of how you customize RAD?
- 1.6. When do you tend to make these customization changes? Do you do this when you first install RAD or do you do this as you go along?
- 1.7. Moving on to a different form of customization, do you continually work within the same workspace or do you periodically create new workspaces, for example, when starting a new project? Why do you choose to do this?
 - 1.7.1. If you use multiple workspaces, how often do you create new ones?
 - 1.7.2. How many do you usually have setup?

2. Perspectives and Views

- 2.1. To get an overall sense of how you use perspectives, from your point of view what is the purpose of perspectives?
- 2.2. Do you customize the layouts of your perspectives? Do you do this when you first install a new version of RAD or do you do this periodically as you go along?
- 2.3. Have you customized any of the menus and toolbars in a perspective? Is this something you do often? I'm trying to get a sense of whether this is a key element of your customization practice or not.
- 2.4. What perspectives do you use? Why do you think that particular set of perspectives works well for you?

- 2.5. Which perspective do you use the most? Within that perspective, which views do you use most frequently?
- 2.6. In terms of switching between perspectives, do you allow RAD to automatically do this sometimes? For example, RAD can automatically flip from the Java perspective to the Debug perspective when you start debugging.
- 2.7. Do you save customized perspectives? If so, what prompts you to save a customized perspective?
- 2.8. Do you reset perspectives? Are you more likely to reset certain perspectives over others?
- 2.9. Fast views are the ones that minimize to an icon and you can access them by clicking on that icon. Do you make use of fast views? Why or why not?
- 2.10. Do you use quick views or in-place views? (e.g., ctrl-O, ctrl-T) Why (and which ones) or why not?
- 2.11. Do you have any general comments about perspectives and views before we move on to the next section of the interview?

3. Capabilities and Roles

The next set of questions is about capabilities and roles, new functionality which has been introduced in the RAD 6.

- 3.1. Are you aware of this new functionality? How would you describe the relationship between roles, capabilities and perspectives? (Do you see their purpose as similar or different?)
- 3.2. To get a sense of your underlying concept of these mechanisms, if you had to come up with a couple of synonyms to describe perspectives, what would you say? How about capabilities? And roles?
- 3.3. What roles do you have enabled? Have you ever changed your roles? If yes, when? If no, why not (do you know how to do so)?
- 3.4. What roles do you identify with during your general development practice? Does that change during the course of a project?
- 3.5. When the dialog comes up asking if you want to allow a capability to be enabled, have you ever chosen *not* to enable it? If yes, why?
- 3.6. Can you give me a ballpark figure of how many times you've enabled a capability in this way?

- 3.7. Also with respect to the capabilities dialog, there is a checkbox that you can click called “Do not ask again”. Have you checked that box? Did you do that the first time the dialog appeared (or when did you decide to do that)?
- 3.8. Do you ever manually enable capabilities as opposed to only enabling them when the capabilities enablement dialog appears? (Do you know how to do so?) What would prompt you to manually enable a capability?
- 3.9. Do you ever disable any capabilities? If yes, what prompts you to do so?
- 3.10. Do you have any other general comments about capabilities or roles before we wrap up this section?

4. Debriefing and Closing Discussion

- 4.1. Do you have any questions or general comments at this point?
- 4.2. Everyone knows that RAD has a complex user interface, and I’m just curious if you have any thoughts about all of this – how perspectives, capabilities and roles might be improved?
- 4.3. If perspectives were absent from this IDE and IBM asked you design something to solve the problem of making a complicated product easier to work with, do you have any thoughts on how this could be done?
- 4.4. I’ve heard from other users that often some of the perspectives are too close to each other and they would choose to trim the set. What do you think? Are there some perspectives, or even roles, that you are too similar and that you would eliminate if you were given the opportunity?
- 4.5. Are you aware of any other products that actively manage user interface complexity that you’d like to comment on? Is (are) their approach(es) better/worse than RAD’s?
- 4.6. Do you think any of these mechanisms – perspectives, roles, and capabilities should be removed from the product?

Appendix B

Layered Interface Study Materials

Includes:

- Background questionnaire
- Post-task questionnaire for Task 1
- Post-task questionnaire for Task 2
- Awareness questionnaire
- Debriefing and interview questions
- Instructions for Simple Task
- Instructions for Complex Task

Background Questionnaire

1. In what age group are you?
 19 and under 30 – 39 50 – 59
 20 – 29 40 – 49 60 +
2. Gender:
 Male
 Female
3. How many hours a week on average do you use a computer (including work and non-work related activities)?
 < 1
 1 - 5
 5 - 10
 > 10
4. Which operating systems do you currently use on a regular basis (at least on a weekly basis)? Please tick all that apply.
 Windows (Microsoft)
 Windows XP
 Windows 2000 or ME
 Windows 95 or 98
 Mac (Apple)
 OS X
 OS 9 or lower
 Linux Please specify distribution(s): _____
5. In terms of computer expertise, would you consider yourself to be:
 Novice
 Intermediate
 Expert
6. What is your current occupation? _____
If student, what are you studying? _____

7. How familiar are you with the following types of computer applications?

	Completely unfamiliar	Mostly unfamiliar	Somewhat familiar	Very familiar
Word processor (e.g., MS Word)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Email (e.g., MS Outlook)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Web browser (e.g., IE, Firefox)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Spreadsheet (e.g., Excel, Lotus 1-2-3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Graphics (e.g., Adobe, Corel Draw)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Presentation software (e.g., PowerPoint)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Database (e.g., MySQL, Oracle)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Music/Video (e.g., iTunes, Quicktime)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Computer games	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other, please specify: _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

8. Do you currently use any Microsoft Office applications on a regular basis?

Yes No

If yes:

a) Which version of Microsoft Office do you use?

- Office 2000
- Office 2002 or XP
- Office 2003 or 2004 for Mac
- Don't know / not aware

b) How familiar are you with the following applications?

	Completely unfamiliar	Mostly unfamiliar	Somewhat familiar	Very familiar
Word	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Excel	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Outlook	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FrontPage	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PowerPoint	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

c) Have you used the beta version of Microsoft Office 2007?

Yes No

9. How did you hear about this study (e.g., poster, email, etc.)? _____

If it was a poster, please specify location: _____

Post-task Questionnaire 1: Travel Task

Please indicate the extent to which you agree or disagree with the statements below.

Use the following scale:

1 ----- 2 ----- 3 ----- 4 ----- 5
Strongly Disagree Disagree Neutral Agree Strongly Agree

- ___ 1. Finding the features I needed in the menus and toolbars was easy.
- ___ 2. I felt overwhelmed by how much “stuff” there was in the menus and toolbars.
- ___ 3. The contents of the menus and toolbars matched my needs.
- ___ 4. It was difficult to navigate through the menus and toolbars.

Post-task Questionnaire 2: Sports Task

Please indicate the extent to which you agree or disagree with the statements below.

Use the following scale:

1 ----- 2 ----- 3 ----- 4 ----- 5
Strongly Disagree Disagree Neutral Agree Strongly Agree

- ___ 1. Finding the features I needed in the menus and toolbars was easy.
- ___ 2. I felt overwhelmed by how much “stuff” there was in the menus and toolbars.
- ___ 3. The contents of the menus and toolbars matched my needs.
- ___ 4. It was difficult to navigate through the menus and toolbars.
- ___ 5. This task (Sports Task) was harder than the first task (Travel Task).
- ___ 6. The skills I learned doing the first task (Travel Task) were helpful in doing this task quickly.
- ___ 7. It was harder to find features during the **first task** (Travel Task) than during the **second task** (Sports Task).


















[For Minimal and Marked conditions only:]

- ___ 8. I preferred the version of the menus and toolbars in the **first task** (Travel Task) to the version used in the **second task** (Sports Task).
- ___ 9. I found it easy to transition from the menus and toolbars used in the first task (Travel Task) to the menus and toolbars used in the second task (Sports Task).

Awareness Questionnaire

For the following list of menu and toolbar features, please indicate whether remember noticing these in the menus and toolbars used for either task.

Note: There may be some features in this list that are *not* found in the menus and toolbars.

	Icon	Type	Feature Name	Yes, I definitely noticed this feature	I vaguely recall seeing this feature	No, I didn't notice this feature
1.		Toolbar	AutoShape Stars and Banners	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.		Menu	New Photo Album	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.		Menu	Package for CD	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.		Toolbar	Invert Picture	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.		Toolbar	Less Contrast	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.		Menu	Hide Slide	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.		Toolbar	Shadow Style	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.		Toolbar	Expand All	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.		Menu	View Color/Grayscale	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.		Toolbar	Set Transparent Color	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11.		Menu	Rehearse Timings	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12.		Menu	Master	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13.		Menu	Play CD Audio Track	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14.		Toolbar	Show Formatting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15.		Menu	Remove Timer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16.		Toolbar	Crop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17.		Menu	View Packaged Layout	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18.		Toolbar	Line Color	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19.		Menu	Assign Slide Manager	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20.		Menu	Insert Slides from Files	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
21.		Toolbar	Insert Diagram or Organization Chart	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
22.		Toolbar	Record Custom Clip	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
23.		Toolbar	Decrease Font Size	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24.		Menu	Alignment	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
25.		Menu	Custom Animation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Debriefing and Interview Questions

Script:

We're interested in making software applications easier to learn and use by organizing features in different ways.

[*For Full condition:*] You used the default version of PowerPoint. Some participants in this study are using other versions, where the menu and toolbar items are organized differently. We're interested in comparing these versions.

[*For Minimal and Marked conditions:*] The first version of PowerPoint you used today was called a [minimal / marked] version, and the second approach was the original PowerPoint version, containing the full set of features in the application.

1. Given that you're not already a PowerPoint user, if you were to start using PowerPoint, which version would you initially prefer to use? Why?

[*If answer is Minimal or Marked:*] Under what circumstances would you want to stay in that version? Or move to the default version?

[*For all conditions:*]

2. If you had the option to switch between the two versions whenever you wanted, do you think you would use both or just one? Why or why not?

3. Do you have any other questions or comments?

Instructions for Simple Task

Of the following set of instructions, 30 steps require generic or basic menu or toolbar commands as described in Chapter 4. For clarity, we include in parentheses the correct command required to complete a step; participants were not given this information.

1. (generic, File > Open) Open the presentation “travel.ppt” from “My Documents\filesForStudy”.
2. (basic, View > Slide Sorter) Display the slides in the Slide Sorter view.
3. Click and drag the second slide so it comes at the beginning of the presentation.
4. (basic, View > Normal) Display the slides in the Normal view again.
5. Make sure that you are on Slide 1.
6. (generic, Formatting > Align Right) (Slide 1) Right align the text “Travel Stories”.
7. Go to Slide 2.
8. (basic, Drawing > Rectangle) (Slide 2) Draw a rectangle over the text “rectangle”.
9. (generic, Formatting > Font Color) (Slide 2) Change the font colour of “Vancouver” to red.
10. (basic, Drawing > Oval) (Slide 2) Draw an oval over the text “oval”.
11. Go to Slide 3.
12. (basic, Format > Slide Layout) (Slide 3) Show the slide layout options.
13. Click the top-right layout under “Text Layouts”.
14. (basic, Drawing > Line) (Slide 3) Draw a line from the “Canada” box to the “Europe” box.
15. (generic, Edit > Cut) (Slide 3) Cut the word “Montreal”.
16. (generic, Edit > Paste) (Slide 3) Paste it before “Vienna”.
17. (basic, Slide Show > View Show) Play the slide show. (Click the mouse to advance through the show.)
18. (basic, Format > Slide Design) (Slide 3) Show the slide design options.
19. (Slide 3) Choose the first design under “Recently Used”.
20. (generic, Edit > Undo) Undo the design template choice.

21. (generic, File > Print) Print the presentation.
22. Go to Slide 4.
23. (basic, Insert > Slide) (Slide 4) Add a new slide to the presentation.
24. (basic, Drawing > Rectangle) (Slide 5) Draw a rectangle that covers the text “Click to add title”.
25. (generic, Help > Microsoft Office PowerPoint Help) Open the PowerPoint help feature.
26. (generic, Insert > Picture from file) (Slide 5) Add the picture “lioness.jpg” from “My Documents \ filesForStudy” to this slide.
27. Go to Slide 2.
28. (basic, Drawing > Arrow) (Slide 2) Draw an arrow from the rectangle to the circle.
29. (generic, Formatting > Italics) (Slide 2) Change the word “Vancouver” to italics.
30. (basic, Drawing > Line) (Slide 2) Draw a line from “Vancouver” to the rectangle.
31. Go to Slide 4.
32. (basic, Drawing > AutoShapes Basic Shapes) (Slide 3) Draw a diamond AutoShape to the left of the sailboat.
33. Go to Slide 3.
34. (basic, Insert > Slide)(Slide 4) Add a new slide to the presentation.
35. Go to Slide 3.
36. (basic, Drawing > Arrow) (Slide 3) Draw an arrow from the word “Cape Town” to the “Canada” box.
37. (generic, Standard > Zoom) (Slide 3) Zoom the view of the slide out to 50%.
38. (basic, Drawing > Oval) (Slide 3) Draw an oval that covers the text “oval”.
39. (basic, Slide Show > View Show) Play the slide show. (Click the mouse to advance through the show.)
40. (basic, Drawing > AutoShapes Block Arrow) (Slide 3) Draw an AutoShape right block arrow from the “Europe” box to the “Africa” box.
41. (generic, Standard > Save) Save the presentation.

Instructions for Complex Task

Of the following set of instructions, 48 steps require generic, basic, or advanced menu or toolbar commands as described in Chapter 4. For clarity, we include in parentheses the correct command required to complete a step; participants were not given this information.

1. (generic, File - Open) Open the presentation “sports.ppt” from “My Documents \ files-ForStudy”.
2. (basic, View - Slide Sorter) Display the slides in the Slide Sorter view.
3. Click and drag the second slide so it comes at the beginning of the presentation.
4. (basic, View > Normal) Display the slides in the Normal view again.
5. (advanced, Standard > Show/Hide Grid) Hide gridlines on the slides.
6. Make sure you are on Slide 1.
7. (advanced, Drawing > Ungroup) (Slide 1) Ungroup the triangle and the oval.
8. (generic, File > Print) (Slide 1) Print the presentation.
9. (advanced, Drawing > Fill Color) (Slide 1) Change the fill colour of the oval to blue.
10. (generic, Formatting > Align Right) (Slide 1) Right align the text “Physical Activity”.
11. Go to Slide 5.
12. (basic, Drawing > Rectangle) (Slide 5) Draw a rectangle over the text “rectangle”.
13. (advanced, Slide Show > Slide Transition) (Slide 5) Change the slide transition to “Blinds Horizontal”.
14. (generic, Formatting > Font Color) (Slide 5) Change the font colour of “Sporting Events Around Town” to blue.
15. (advanced, Drawing > Order - Send to back) (Slide 5) Put the blue diamond behind the triangle in the order of images on the slide.
16. (basic, Drawing > Oval) (Slide 5) Draw an oval over the text “oval”.
17. Go to Slide 2.
18. (advanced, Format > Line Spacing) (Slide 2) Click on the line “Improve your health” and change the line spacing to '2' lines.

19. Go to Slide 4.
20. (basic, Format > Slide Layout) (Slide 4) Show the slide layout options.
21. (Slide 4) Click the top-right layout under “Text Layouts”.
22. (advanced, Drawing > Dash Style) (Slide 4) Change the dash style of the thick black line to “Square dot”.
23. (basic, Drawing > Line) (Slide 4) Draw an line from the “Swimming” box to the “Water polo” box.
24. (advanced, Picture > Recolor Picture) (Slide 4) Recolor the picture of the schoolbus. Once the options are open
25. (generic, Edit > Cut) (Slide 4) Cut the word “Baseball”.
26. (generic, Edit > Paste) (Slide 4) Paste it after “Track and Field”.
27. Go to Slide 2.
28. (advanced, Tools > Language) (Slide 2) First select the text “Je fais de la natation”
29. (basic, Slide Show > View Show) Play the slide show. (Click the mouse to advance through the show.)
30. (advanced, Picture > Picture Color) (Slide 2) Change the color of the soccer picture to “Automatic”.
31. (basic, Format > Slide Design) (Slide 2) Show the slide design options.
32. (Slide 2) Choose the first design under “Recently Used”.
33. (generic, Edit > Undo) Undo the design template choice.
34. Go to Slide 5.
35. (basic, Insert > new slide) (Slide 5) Add a new slide to the presentation.
36. Make sure you are on Slide 6.
37. (generic, Help > Microsoft Office PowerPoint Help) Open the Page Setup
38. (generic, Insert > Picture from file) (Slide 6) Add the picture “running.jpg” from “My Documents \ filesForStudy” to this slide.
39. (basic, Drawing > Arrow) (Slide 3) Draw an arrow from the rectangle to the triangle.

40. (generic, Formatting > Italics) (Slide 3) Change the word “Hiking” to italics.
41. (basic, Drawing > Line) (Slide 3) Draw a line from “Hiking” to the rectangle.
42. (advanced, Format > Background) (Slide 3) Change the slide background colour to white.
43. Go to Slide 5.
44. (basic, Insert > New Slide) (Slide 5) Add a new slide to the presentation.
45. (Slide 6) Type the word “soccer” where it says “Click to add text”.
46. (advanced, Insert > Duplicate slide) (Slide 6) Create a duplicate slide.
47. (advanced, Insert > Sound from clip) (Slide 7) Add the sound “Claps Cheers” from the sound clip organizer. (When asked how you want it to play
48. Go to Slide 4.
49. (basic, Drawing > AutoShapes Basic Shapes) (Slide 4) Draw a hexagon AutoShape that covers the text “hexagon”.
50. Go to Slide 3.
51. (basic, Drawing > Arrow) (Slide 3) Draw an arrow from the “Hiking” to “Kayaking”.
52. (generic, Standard > Zoom) (Slide 3) Zoom the view of the slide out to 50%.
53. (basic, Drawing > Oval) (Slide 3) Draw an oval that covers the text “oval”.
54. (advanced, Slide Show > Set Up Show) Open the Set Up Show dialog. Once it is open
55. (advanced, View > Grid and Guides) Show the settings for grids and guides. Once it is open
56. (basic, Slide Show > View Show) Play the slide show. (Click the mouse to advance through the show.)
57. Go to Slide 5.
58. (basic, Drawing > AutoShapes Block Arrow) (Slide 5) Draw an AutoShape down block arrow from the triangle to “Sun Run”.
59. Go to Slide 6.
60. (basic, Drawing > Rectangle) (Slide 6) Draw a rectangle that covers the text “Click to add title”.

61. (advanced, Slide Show > Animation Schemes) (Slide 6) Show the animation schemes options.
62. Select the first animation scheme under “Recently Used”.
63. (advanced, Edit > Delete Slide) (Slide 7) Delete this slide from the presentation.
64. (advanced, View > Notes Page) (Slide 7) Show the notes page for this slide.
65. (generic, Standard > Save) Save the presentation.

Appendix C

Screen Size Study Materials

Includes:

- Background questionnaire
- Example sheet for introducing awareness test
- End of condition questionnaire
- End of condition awareness test
- End of study questionnaire

Background Questionnaire

1. In what age group are you?
 19 and under 30 – 39 50 – 59
 20 – 29 40 – 49 60 +

2. Gender:
 Male
 Female

3. Handedness:
 Left-handed
 Right-handed

4. How many hours a week on average do you use a computer (including work and non-work related activities)?
 < 1
 1 - 5
 6 - 10
 > 10

5. Which operating systems do you currently use on a regular basis (at least on a weekly basis)? Please tick all that apply.
 Windows (Microsoft)
 Windows XP
 Windows 2000 or ME
 Windows 95 or 98
 Mac (Apple)
 OS X
 OS 9 or lower
 Linux Please specify: _____
 Other Please specify: _____

6. In terms of computer expertise, would you consider yourself to be:
 Novice
 Intermediate
 Expert

7. What is your current occupation? _____

If student, what are you studying? _____

8. How familiar are you with using the following types of computer applications?

	Completely unfamiliar	Mostly unfamiliar	Somewhat familiar	Very familiar
Word processor (e.g., MS Word)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Email (e.g., MS Outlook)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Web browser (e.g., IE, Firefox)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Spreadsheet (e.g., Excel, Lotus 1-2-3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Graphics (e.g., Adobe, Corel Draw)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Presentation software (e.g., PowerPoint)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Database (e.g., MySQL, Oracle)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Music/Video (e.g., iTunes, Quicktime)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Computer games	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other, please specify: _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

9. How familiar are you with using the following types of devices?

	Completely unfamiliar	Mostly unfamiliar	Somewhat familiar	Very familiar
MP3 player (e.g., iPod)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Personal Digital Assistant - PDA (e.g., Blackberry, Palm Pilot)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cell phone	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Introduction of Awareness Example Test

Note: There may be some items here that were *not* found on the list.

	Menu item	Yes, I definitely saw this item	I vaguely recall seeing this item	No, I didn't see this item
1.	Loveseat	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	Windsor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	Lemon	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	Hamilton	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	Begonia	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	Chair	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	Edmonton	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	Peach	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	Daffodil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Tulip	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11.	Sofa	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12.	Whitehorse	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

End of Condition Questionnaire

Please indicate the extent to which you agree or disagree with the statements below by circling the appropriate number on each scale.

The menus were difficult to use.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

I could efficiently and quickly find items in the menus.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

It was hard to learn the full set of menu items.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

The menus were satisfying to use.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

[For adaptive conditions only]

The items in the top part of the menus remained relatively consistent from one selection to the next.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

The items in the top part of the menus were unpredictable.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

(Sample) End of Condition Awareness Test

Note: There may be some items in this list that are *not* found in the menus.

	Menu item	Yes, I definitely saw this item	I vaguely recall seeing this item	No, I didn't see this item
1.	Gondola	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	Mozart	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	Basketball	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	Ballpoint	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	Crayon	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	Grape	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	Mazda	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	Bigfoot	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	Stool	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Tiger	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11.	Amazon	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12.	Paris	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13.	Orange	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14.	London	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15.	Terrier	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16.	Chopin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17.	Adidas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18.	Roman	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Appendix D

New Task Study Materials

Includes:

- Background questionnaire
- Sample end of condition awareness test
- Sample end of condition subjective measures questionnaire
- End of study questionnaire

Background Questionnaire

1. What is your age? _____

2. Gender:
 Male Female

3. Handedness:
 Left-handed Right-handed

4. Do you use a computer for work? (either at home or work).
 Yes No N/A

If yes, on an average day (or week), approximately how many hours do you spend using a computer for this?
_____ hours per Day Week

5. Do you use a computer for leisure or personal tasks? (either at home or work).
 Yes No N/A

If yes, on an average day (or week), approximately how many hours do you spend using a computer for this?
_____ hours per Day Week

6. What kinds of computers have you used? Tick all that apply.
 PC (Windows)
 PC (Linux)
 Mac/Apple
 Unix
 Laptop/Notebook
 Tablet
 Handheld (PDA/PalmPilot/Blackberry)
 Other _____

7. In terms of computer expertise, would you consider yourself to be:
 Novice Intermediate Expert

8. What is your current occupation? _____
If **student**, what are you studying? _____

9. How familiar are you with using the following types of computer applications?

	Completely unfamiliar	Mostly unfamiliar	Somewhat familiar	Very familiar
Word processor (e.g., MS Word)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Email (e.g., MS Outlook)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Web browser (e.g., IE, Firefox)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Spreadsheet (e.g., Excel, Lotus 1-2-3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Graphics (e.g., Adobe, Corel Draw)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Presentation software (e.g., PowerPoint)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Database (e.g., MySQL, Oracle)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Music/Video (e.g., iTunes, Quicktime)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Computer games	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other, please specify: _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(Sample) End of Condition Awareness Test

Note: There may be some items in this list that are *not* found in the menus.

	Menu item	Yes, I definitely saw this item	I vaguely recall seeing this item	No, I didn't see this item
1.	Lasagna	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	Prance	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	Backgammon	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	Toonie	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	China	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	Poodle	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	Labatt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	Sedan	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	Hatchback	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Scrabble	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11.	Mountain	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12.	Carrot	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13.	Singapore	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14.	Ballroom	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15.	Flamenco	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16.	Japan	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17.	Cabernet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18.	Mongol	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(Sample) End of Condition Questionnaire

Please indicate the extent to which you agree or disagree with the statements below by circling the appropriate number on each scale.

The menus were difficult to use.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

I could efficiently and quickly find items in the menus.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

It was hard to learn the full set of menu items.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

The menus were satisfying to use.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

It was hard to find items I selected infrequently.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

It was easy to remember words that were in the menus but that I didn't select.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

End of Study Questionnaire

For each of the 3 types of menu (S, T1, T2), please indicate the extent to which you agree or disagree with the statements below by circling and labelling the corresponding number. For example:

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

Labels:

S = menus never changed

T1 = first menus with a separate top area

T2 = second menus with a separate top area

The menus were difficult to use.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

I could efficiently and quickly find items in the menus.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

It was hard to learn the full set of menu items.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

The menus were satisfying to use.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

It was hard to find items I selected infrequently.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

It was easy to remember words that were in the menus but that I didn't select.

1 ----- 2 ----- 3 ----- 4 ----- 5 ----- 6 ----- 7
Disagree Neutral Agree

Which type of menu did you prefer over the others? (S, T1 or T2) _____

Why?

Appendix E

Ephemeral Adaptation Study Materials

Includes:

- Background questionnaire, Studies 1 and 2
- End of condition questionnaire, Study 1
- End of condition questionnaire, Study 2
- End of study questionnaire, Studies 1 and 2

Ephemeral Adaptation Studies 1 and 2 : Background Questionnaire

1. What is your age? _____

2. Gender:
 Male Female

3. Handedness:
 Left-handed Right-handed

4. Do you use a computer for work? (either at home or work).
 Yes No N/A

If yes, on an average day (or week), approximately how many hours do you spend using a computer for this?
_____ hours per Day Week

5. Do you use a computer for leisure or personal tasks? (either at home or work).
 Yes No N/A

If yes, on an average day (or week), approximately how many hours do you spend using a computer for this?
_____ hours per Day Week

6. What kinds of computers have you used? Tick all that apply.
 PC (Windows)
 PC (Linux)
 Mac/Apple
 Unix
 Laptop/Notebook
 Tablet
 Handheld (PDA/PalmPilot/Blackberry)
 Other _____

7. In terms of computer expertise, would you consider yourself to be:
 Novice Intermediate Expert

8. What is your current occupation? _____
If **student**, what are you studying? _____

9. How familiar are you with using the following types of computer applications?

	Completely unfamiliar	Mostly unfamiliar	Somewhat familiar	Very familiar
Word processor (e.g., MS Word)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Email (e.g., MS Outlook)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Web browser (e.g., IE, Firefox)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Spreadsheet (e.g., Excel, Lotus 1-2-3)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Graphics (e.g., Adobe, Corel Draw)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Presentation software (e.g., PowerPoint)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Database (e.g., MySQL, Oracle)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Music/Video (e.g., iTunes, Quicktime)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Computer games	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other, please specify: _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Ephemeral Adaptation Study 1: End of Condition Questionnaire

Please indicate the extent to which you agree or disagree with the statements below by circling the appropriate number on each scale.

The menus were difficult to use.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

I could efficiently and quickly find items in the menus.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

The menus were frustrating to use.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

The menus were satisfying to use.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

If these were adaptive menus:

Were you able to distinguish the items that appeared before others when you opened a menu? (circle one)

Yes No

If yes:

This behaviour helped me find items more quickly.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

I found this behaviour distracting.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

Ephemeral Adaptation Study 2: End of Condition Questionnaire

Please indicate the extent to which you agree or disagree with the statements below by circling the appropriate number on each scale.

The menus were difficult to use.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

I could efficiently and quickly find items in the menus.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

The menus were frustrating to use.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

The menus were satisfying to use.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

[For adaptive conditions only]

In this condition, some items appeared before others when you opened a menu.

It was easy to distinguish these items.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

This behaviour helped me find items more quickly.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

I found this behaviour distracting.

1 - - - - 2 - - - - 3 - - - - 4 - - - - 5 - - - - 6 - - - - 7
Disagree Neutral Agree

Ephemeral Adaptation Studies 1 and 2: End of Study Questionnaire

You used 3 types of menus today. Please order them for each of the following, using the labels 1 to 3 for each of the menu types.

Fastest ___ ___ ___ Slowest

Why?

Easiest ___ ___ ___ Most difficult

Why?

Least frustrating ___ ___ ___ Most frustrating

Why?

Most preferred ___ ___ ___ Least preferred

Why?

Did any of the menu types seem similar to you? Yes No

If yes, which ones?

Why?

Do you have any additional comments?

Appendix F

UBC Research Ethics Board Certificates



Certificate of Approval

PRINCIPAL INVESTIGATOR McGrenere, J	DEPARTMENT Computer Science	NUMBER B03-0144
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT UBC Campus ,		
CO-INVESTIGATORS: Bunt, Andrea, Science; Conati, Cristina, Computer Science; Findlater, Leah, Computer Science		
SPONSORING AGENCIES Natural Science Engineering Research Council		
TITLE: Adaptive and Adaptable Information Technology		
APPROVAL DATE 05-10-13 <small>(yr/mo/day)</small>	TERM (YEARS) 1	AMENDMENT: Nov. 17, 2005, Advertisement / Consent form / Questionnaires
		AMENDMENT APPROVED: NOV 24 2005
<p style="text-align: center;">CERTIFICATION:</p> <p style="text-align: center;">The protocol describing the above-named project has been reviewed by the Committee and the experimental procedures were found to be acceptable on ethical grounds for research involving human subjects.</p> <div style="text-align: center; margin: 20px 0;"> </div> <p style="text-align: center;"><i>Approved on behalf of the Behavioural Research</i> <i>by one of the following:</i> Dr. Peter Suedfeld, Chair, Dr. Susan Rowley, Associate Chair</p> <p style="text-align: center;">This Certificate of Approval is valid for the above term provided there is no change in the experimental procedures</p>		



Certificate of Approval

PRINCIPAL INVESTIGATOR McGrener, J		DEPARTMENT Computer Science		NUMBER B03-0144	
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT UBC Campus ,					
CO-INVESTIGATORS: Bunt, Andrea, Science; Conati, Cristina, Computer Science; Findlater, Leah, Computer Science					
SPONSORING AGENCIES Natural Science Engineering Research Council					
TITLE: Adaptive and Adaptable Information Technology					
APPROVAL DATE 05-10-13 <small>(y/mo/day)</small>		TERM (YEARS) 1	AMENDMENT: Mar. 27, 2006, Reimbursement / Recruitment letter / Online study documents		AMENDMENT APPROVED: MAR 28 2006
<p>CERTIFICATION:</p> <p style="text-align: center;">The request for continuing review of an amendment to the above-named project has been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.</p> <div style="text-align: center; margin: 10px 0;"> </div> <p style="text-align: center;"><i>Approved on behalf of the Behavioural Research Ethics Board</i> by one of the following:</p> <ul style="list-style-type: none"> Dr. Peter Suedfeld, Chair, Dr. Susan Rowley, Associate Chair Dr. Jim Rupert, Associate Chair Dr. Arminee Kazanjian, Associate Chair <p style="text-align: center;">This Certificate of Approval is valid for the above term provided there is no change in the experimental procedures</p>					



Certificate of Approval

PRINCIPAL INVESTIGATOR McGrenere, J	DEPARTMENT Computer Science	NUMBER B03-0144
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT UBC Campus ,		
CO-INVESTIGATORS: Bunt, Andrea, Science; Conati, Cristina, Computer Science; Findlater, Leah, Computer Science		
SPONSORING AGENCIES Natural Science Engineering Research Council		
TITLE: Adaptive and Adaptable Information Technology		
APPROVAL DATE 05-10-13 <small>(yr/mo/day)</small>	TERM (YEARS) 1	AMENDMENT: Apr. 4, 2006, Reimbursement / Recruitment letter / Online study documents
AMENDMENT APPROVED: APR 13 2006		
<p>CERTIFICATION:</p> <p>The request for continuing review of an amendment to the above-named project has been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.</p> <div style="text-align: center; margin: 20px 0;"> </div> <p style="text-align: center;"><i>Approved on behalf of the Behavioural Research Ethics Board</i> <i>by one of the following:</i></p> <p style="text-align: center;"> Dr. Peter Suedfeld, Chair, Dr. Susan Rowley, Associate Chair Dr. Jim Rupert, Associate Chair Dr. Arminee Kazanjian, Associate Chair </p> <p>This Certificate of Approval is valid for the above term provided there is no change in the experimental procedures</p>		



Certificate of Approval

PRINCIPAL INVESTIGATOR McGrener, J	DEPARTMENT Computer Science	NUMBER B03-0144	
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT UBC Campus ,			
CO-INVESTIGATORS: Bunt, Andrea, Science; Conati, Cristina, Computer Science; Findlater, Leah, Computer Science			
SPONSORING AGENCIES Natural Science Engineering Research Council			
TITLE: Adaptive and Adaptable Information Technology			
APPROVAL DATE 05-10-13 <small>(y/m/d)</small>	TERM (YEARS) 1	AMENDMENT: Aug. 17, 2006, Questionnaires / Aug. 19, 2006, Consent form / Aug. 21, 2006, Recruitment method / Recruitment letter / Poster / Tests	AMENDMENT APPROVED: AUG 25 2006
<p>CERTIFICATION:</p> <p>The request for continuing review of an amendment to the above-named project has been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.</p> <div style="text-align: center; margin: 10px 0;"> </div> <p style="text-align: center;"><i>Approved on behalf of the Behavioural Research Ethics Board</i> by one of the following: Dr. Peter Suedfeld, Chair, Dr. Jim Rupert, Associate Chair Dr. Arminee Kazanjian, Associate Chair</p> <p>This Certificate of Approval is valid for the above term provided there is no change in the experimental procedures</p>			



The University of British Columbia
 Office of Research Services
Behavioural Research Ethics Board
 Suite 102, 6190 Agronomy Road,
 Vancouver, B.C. V6T 1Z3

CERTIFICATE OF APPROVAL - MINIMAL RISK AMENDMENT

PRINCIPAL INVESTIGATOR: Joanna McGrenere	DEPARTMENT: UBC/Science/Computer Science	UBC BREB NUMBER: H03-80144
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT:		
Institution	Site	
UBC	Point Grey Site	
Other locations where the research will be conducted: N/A		
CO-INVESTIGATOR(S): Andrea Bunt Cristina Conati Leah Findlater		
SPONSORING AGENCIES: Natural Sciences and Engineering Research Council of Canada (NSERC) - "Orsil title - Network for effective collaboratin technologies through advanced research" - "Design and evaluation of adaptive and adaptable information technology"		
PROJECT TITLE: Adaptive and adaptable information technology		

Expiry Date - Approval of an amendment does not change the expiry date on the current UBC BREB approval of this study. An application for renewal is required on or before: October 26, 2007

AMENDMENT(S):	AMENDMENT APPROVAL DATE:
----------------------	---------------------------------

April 25, 2007

N/A

The amendment(s) and the document(s) listed above have been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.

*Approval is issued on behalf of the Behavioural Research Ethics Board
and signed electronically by one of the following:*

Dr. Peter Suedfeld, Chair
Dr. Jim Rupert, Associate Chair
Dr. Arminee Kazanjian, Associate Chair
Dr. M. Judith Lynam, Associate Chair
Dr. Laurie Ford, Associate Chair



The University of British Columbia
 Office of Research Services
Behavioural Research Ethics Board
 Suite 102, 6190 Agronomy Road,
 Vancouver, B.C. V6T 1Z3

CERTIFICATE OF APPROVAL - MINIMAL RISK AMENDMENT

PRINCIPAL INVESTIGATOR: Joanna McGrenere	DEPARTMENT: UBC/Science/Computer Science	UBC BREB NUMBER: H03-80144
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT:		
Institution	Site	
UBC	Point Grey Site	
Other locations where the research will be conducted: N/A		
CO-INVESTIGATOR(S): Andrea Bunt Cristina Conati Leah Findlater		
SPONSORING AGENCIES: Natural Sciences and Engineering Research Council of Canada (NSERC) - "Design and evaluation of adaptive and adaptable information technology" - "Orsil title - Network for effective collaboratin technologies through advanced research"		
PROJECT TITLE: Adaptive and adaptable information technology		

Expiry Date - Approval of an amendment does not change the expiry date on the current UBC BREB approval of this study. An application for renewal is required on or before: October 26, 2007

AMENDMENT(S):	AMENDMENT APPROVAL DATE:
----------------------	---------------------------------

June 19, 2007

Document Name	Version	Date
Consent Forms:		
June 2007 consent	N/A	June 15, 2007
Advertisements:		
June 2007 poster	N/A	June 15, 2007
Questionnaire, Questionnaire Cover Letter, Tests:		
June 2007 questionnaires	N/A	June 15, 2007
Other Documents:		
Digit Symbol Substitution Test	N/A	May 15, 2003

The amendment(s) and the document(s) listed above have been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.

*Approval is issued on behalf of the Behavioural Research Ethics Board
and signed electronically by one of the following:*

Dr. Peter Suedfeld, Chair
Dr. Jim Rupert, Associate Chair
Dr. Arminee Kazanjian, Associate Chair
Dr. M. Judith Lynam, Associate Chair
Dr. Laurie Ford, Associate Chair



The University of British Columbia
 Office of Research Services
Behavioural Research Ethics Board
 Suite 102, 6190 Agronomy Road,
 Vancouver, B.C. V6T 1Z3

CERTIFICATE OF APPROVAL - AMENDMENT & RENEWAL

PRINCIPAL INVESTIGATOR: Joanna McGreener	DEPARTMENT: UBC/Science/Computer Science	UBC BREB NUMBER: H03-80144
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT:		
Institution	Site	
UBC	Vancouver (excludes UBC Hospital)	
Other locations where the research will be conducted: N/A		
CO-INVESTIGATOR(S): Cristina Conati Leah Findlater		
SPONSORING AGENCIES: Natural Sciences and Engineering Research Council of Canada (NSERC) - "Orsil title - Network for effective collaboratin technologies through advanced research" - "Design and evaluation of adaptive and adaptable information technology"		
PROJECT TITLE: Adaptive and adaptable information technology		

CERTIFICATE EXPIRY DATE: October 15, 2008

AMENDMENT(S):	RENEWAL AND AMENDMENT APPROVAL DATE: October 15, 2007
Document Name	Version Date

The application for continuing ethical review and the amendment(s) for the above-named project have been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.

Approval is issued on behalf of the Behavioural Research Ethics Board



The University of British Columbia
 Office of Research Services
Behavioural Research Ethics Board
 Suite 102, 6190 Agronomy Road,
 Vancouver, B.C. V6T 1Z3

CERTIFICATE OF APPROVAL - MINIMAL RISK AMENDMENT

PRINCIPAL INVESTIGATOR: Joanna McGreener	DEPARTMENT: UBC/Science/Computer Science	UBC BREB NUMBER: H03-80144
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT:		
Institution	Site	
UBC	Vancouver (excludes UBC Hospital)	
Other locations where the research will be conducted: N/A		
CO-INVESTIGATOR(S): Cristina Conati Leah Findlater		
SPONSORING AGENCIES: Natural Sciences and Engineering Research Council of Canada (NSERC) - "Orsil title - Network for effective collaboratin technologies through advanced research" - "Design and evaluation of adaptive and adaptable information technology"		
PROJECT TITLE: Adaptive and adaptable information technology		

Expiry Date - Approval of an amendment does not change the expiry date on the current UBC BREB approval of this study. An application for renewal is required on or before: October 15, 2008

AMENDMENT(S):	AMENDMENT APPROVAL DATE: December 3, 2007
----------------------	---

Document Name	Version	Date
Advertisements:		
November 2007 poster	N/A	November 22, 2007
Questionnaire, Questionnaire Cover Letter, Tests:		
November 2007 short background questionnaire	N/A	November 22, 2007
The amendment(s) and the document(s) listed above have been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.		
<p><i>Approval is issued on behalf of the Behavioural Research Ethics Board and signed electronically by one of the following:</i></p> <hr/> <p>Dr. M. Judith Lynam, Chair Dr. Jim Rupert, Associate Chair r. Laurie Ford, Associate Chair</p>		



The University of British Columbia
Office of Research Services
Behavioural Research Ethics Board
Suite 102, 6190 Agronomy Road,
Vancouver, B.C. V6T 1Z3

CERTIFICATE OF APPROVAL - MINIMAL RISK AMENDMENT

PRINCIPAL INVESTIGATOR: Joanna McGrenere	DEPARTMENT: UBC/Science/Computer Science	UBC BREB NUMBER: H03-80144
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT:		
Institution	Site	
UBC Other locations where the research will be conducted: N/A	Vancouver (excludes UBC Hospital)	

CO-INVESTIGATOR(S): Cristina Conati Leah Findlater
SPONSORING AGENCIES: Natural Sciences and Engineering Research Council of Canada (NSERC) - "Orsil title - Network for effective collaboratin technologies through advanced research" - "Design and evaluation of adaptive and adaptable information technology" UBC Dean of Science
PROJECT TITLE: Adaptive and adaptable information technology

Expiry Date - Approval of an amendment does not change the expiry date on the current UBC BREB approval of this study. An application for renewal is required on or before: October 15, 2008

AMENDMENT(S):	AMENDMENT APPROVAL DATE: March 27, 2008	
Document Name	Version	Date
The amendment(s) and the document(s) listed above have been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.		
<p><i>Approval is issued on behalf of the Behavioural Research Ethics Board</i> Dr. M. Judith Lynam, Chair Dr. Ken Craig, Chair Dr. Jim Rupert, Associate Chair Dr. Laurie Ford, Associate Chair Dr. Daniel Salhani, Associate Chair Dr. Anita Ho, Associate Chair</p>		



The University of British Columbia
 Office of Research Services
Behavioural Research Ethics Board
 Suite 102, 6190 Agronomy Road,
 Vancouver, B.C. V6T 1Z3

CERTIFICATE OF APPROVAL - MINIMAL RISK AMENDMENT

PRINCIPAL INVESTIGATOR: Joanna McGrenere	DEPARTMENT: UBC/Science/Computer Science	UBC BREB NUMBER: H03-80144
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT:		
Institution	Site	
UBC	Vancouver (excludes UBC Hospital)	
Other locations where the research will be conducted: N/A		
CO-INVESTIGATOR(S): Cristina Conati Leah Findlater		
SPONSORING AGENCIES: Natural Sciences and Engineering Research Council of Canada (NSERC) - "Design and evaluation of adaptive and adaptable information technology" - "Orsil title - Network for effective collaboratin technologies through advanced research" UBC Dean of Science		
PROJECT TITLE: Adaptive and adaptable information technology		

Expiry Date - Approval of an amendment does not change the expiry date on the current UBC BREB approval of this study. An application for renewal is required on or before: October 15, 2008

AMENDMENT(S):	AMENDMENT APPROVAL DATE:
----------------------	---------------------------------

June 10, 2008

Document Name	Version	Date
---------------	---------	------

The amendment(s) and the document(s) listed above have been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.

Approval is issued on behalf of the Behavioural Research Ethics Board

Dr. M. Judith Lynam, Chair

Dr. Ken Craig, Chair

Dr. Jim Rupert, Associate Chair

Dr. Laurie Ford, Associate Chair

Dr. Daniel Salhani, Associate Chair

Dr. Anita Ho, Associate Chair



The University of British Columbia
 Office of Research Services
Behavioural Research Ethics Board
 Suite 102, 6190 Agronomy Road,
 Vancouver, B.C. V6T 1Z3

CERTIFICATE OF APPROVAL - MINIMAL RISK AMENDMENT

PRINCIPAL INVESTIGATOR: Joanna McGrenere	DEPARTMENT: UBC/Science/Computer Science	UBC BREB NUMBER: H03-80144
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT:		
Institution	Site	
UBC	Vancouver (excludes UBC Hospital)	
Other locations where the research will be conducted: N/A		
CO-INVESTIGATOR(S): Cristina Conati Leah Findlater		
SPONSORING AGENCIES: Natural Sciences and Engineering Research Council of Canada (NSERC) - "Orsil title - Network for effective collaboratin technologies through advanced research" - "Design and evaluation of adaptive and adaptable information technology" UBC Dean of Science		
PROJECT TITLE: Adaptive and adaptable information technology		

Expiry Date - Approval of an amendment does not change the expiry date on the current UBC BREB approval of this study. An application for renewal is required on or before: October 15, 2008

AMENDMENT(S):	AMENDMENT APPROVAL DATE:
----------------------	---------------------------------

August 25, 2008

Document Name	Version	Date
---------------	---------	------

The amendment(s) and the document(s) listed above have been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.

Approval is issued on behalf of the Behavioural Research Ethics Board

Dr. M. Judith Lynam, Chair

Dr. Ken Craig, Chair

Dr. Jim Rupert, Associate Chair

Dr. Laurie Ford, Associate Chair

Dr. Daniel Salhani, Associate Chair

Dr. Anita Ho, Associate Chair



The University of British Columbia
 Office of Research Services
Behavioural Research Ethics Board
 Suite 102, 6190 Agronomy Road,
 Vancouver, B.C. V6T 1Z3

CERTIFICATE OF APPROVAL- MINIMAL RISK RENEWAL

PRINCIPAL INVESTIGATOR: Joanna McGrener	DEPARTMENT: UBC/Science/Computer Science	UBC BREB NUMBER: H03-80144
INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT:		
Institution	Site	
UBC	Vancouver (excludes UBC Hospital)	
Other locations where the research will be conducted: N/A		
CO-INVESTIGATOR(S): Cristina Conati Leah Findlater		
SPONSORING AGENCIES: Natural Sciences and Engineering Research Council of Canada (NSERC) - "Orsil title - Network for effective collaboratin technologies through advanced research" - "Design and evaluation of adaptive and adaptable information technology" UBC Dean of Science		
PROJECT TITLE: Adaptive and adaptable information technology		

EXPIRY DATE OF THIS APPROVAL: October 15, 2009

APPROVAL DATE: October 15, 2008

The Annual Renewal for Study have been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.

Approval is issued on behalf of the Behavioural Research Ethics Board

Dr. M. Judith Lynam, Chair
Dr. Ken Craig, Chair
Dr. Jim Rupert, Associate Chair
Dr. Laurie Ford, Associate Chair
Dr. Daniel Salhani, Associate Chair
Dr. Anita Ho, Associate Chair