Exploring the Design Space for Adaptive Graphical User Interfaces

Krzysztof Z. Gajos[1,2], Mary Czerwinski[1], Desney S. Tan[1] and Daniel S. Weld[2]

[1] Microsoft Research
One Microsoft Way, Redmond, WA 98052
{marycz,desney}@microsoft.com

[2] University of Washington
Box 352350, Seattle, WA 98195
{kgajos,weld}@cs.washington.edu

ABSTRACT

For decades, researchers have presented different adaptive user interfaces and discussed the pros and cons of adaptation on task performance and satisfaction. Little research, however, has been directed at isolating and understanding those aspects of adaptive interfaces which make some of them successful and others not. We have designed and implemented three adaptive graphical interfaces and evaluated them in two experiments along with a non-adaptive baseline. In this paper we synthesize our results with previous work and discuss how different design choices and interactions affect the success of adaptive graphical user interfaces.

Author Keywords

Adaptive interfaces, user study

ACM Classification Keywords

H5.2 Information Interfaces and Presentation: User Interfaces – *Interactions Styles, Evaluation/Methodology*

INTRODUCTION

Automatic adaptation of user interfaces has been discussed for more than two decades. Surprisingly, however, there appear to be few experimental results that systematically evaluate the space of designs in a manner which informs the discussion. The few studies that have been published show examples of both successful and unsuccessful adaptation methods, but do not comprehensively consider the reasons underlying this success or failure.

We believe that it is important to understand the reasons that make some adaptive interfaces effective and pleasing to use while others are a frustrating impediment. Understanding these interfaces through strong empirical or theoretical studies is particularly important, since adaptive interfaces are now being introduced into mainstream productivity software (e.g., Microsoft's Smart Menus™ and the Windows XP Start Menu), and used by an ever increasing number of people.

While the past work often relied on theoretically possible benefits of any particular adaptation design to try to predict its adoption by the user [4,9,14], we turn to the notion of subjective benefits and costs of adaptation as perceived by the user and we try to identify the design choices that influence these perceptions.

We make two contributions in this paper. First, we present two experiments in which we compare three adaptive user interfaces to a non-adaptive baseline. We observed mixed results with these interfaces, which allows us to effectively compare the relative importance of various dimensions within the design space. Secondly, we analyze these (and past) results and point out those design choices, which clearly affect the success of different adaptive interfaces. We conclude by suggesting several promising directions for future research.

PREVIOUS WORK

The first rigorous study of adaptation was reported in 1985, when Greenberg and Witten demonstrated a successful adaptive interface for a menu-driven application [4]. In their study, as in most others, users were novices on the task and the interface, and long-term effects were not studied. This is important, because the study design precluded motor memory or expertise with the interface from being an issue.

In 1989 Mitchell and Shneiderman [9] provided one of the first strong negative results: a (different) menu-driven interface, which adapted by reordering elements in the menu based on their relative frequencies of use. While this reordering could *plausibly* result in improved performance, users had reduced performance in practice, reported being disoriented by the changing nature of the interface, and expressed a strong preference for the static menus, where items were listed in the alphabetical order.

In 1994 Sears and Shneiderman demonstrated that another approach to adapting menus, called Split Menus, resulted in improved performance and user satisfaction [11]. However, the interface was adapted only once per user in a long-term study and once per session in a performance study. It is important to note that their original Split Menu design caused promoted elements to be *moved* rather than copied to the top of the menu; they were no longer available in their original locations. Some commercially deployed versions of Split Menus changed this behavior so that promoted items were available both in the original location and at the top of the interface.

Although a number of adaptive systems were soon developed, the next rigorous studies of adaptation were reported only recently. In 2002, McGrenere et al. reported a long-term study comparing the “out of the box” interface shipped with Microsoft Word to a very sparse one, which each user customized for his or her needs [8]. Additionally, the default interface had the Microsoft Smart Menus adaptation turned on, while the customized version had that adaptation turned off. The study showed that users appreciated having the personalized interface and that they used it more than the factory-supplied version. The authors interpreted the results as evidence against using adaptation, although there were large functionality differences between the two UI designs. Therefore, it is difficult to tell if the main effects of their study were due to the difference in the complexity of the two interfaces, or to the fact that one of them exhibited adaptive behavior.

In 2004 Findlater and McGrenere conducted a laboratory study to compare static, customizable and adaptive versions of Split Menus (using the original design where items are *moved* to the top location) [2]. They found that users generally preferred the customizable version to the adaptive menus. In terms of performance, adaptive menus were not faster than either of the two other conditions. From this study, they concluded that user-driven customization is a much more viable approach for personalizing UIs than system-driven automatic adaptation.

In 2005, Tsandilas and Shraefel reported on a study which, rather than evaluating particular adaptive interfaces, instead focused on the impact of accuracy of the adaptive algorithm on users’ performance and satisfaction. They compared two different adaptive interfaces: the baseline, where suggested menu items were highlighted, and shrinking interface, which also reduced the font size of non-suggested elements [13].

Even more recently, Gajos et al. reported on a preliminary user study comparing two adaptive interfaces for a (software) graphing calculator interface to their corresponding non-adaptive baselines [3]. In their S*plit* interface, most of the calculator’s functionality was placed in a two-level menu and the frequently used groups of functions were dynamically duplicated to a specially designated part of the top-level interface. In the *Altered Prominence* interface, all functionality was available at the top level of the interface, thus making it very busy. The frequently used groups of functions were highlighted for easier visual search. The study showed user preference for the split interface over the corresponding non-adaptive baseline, while some users expressed strong dislike for the Altered Prominence interface.

Research on user-driven customization has shown that users often fail to customize (e.g. [10]) and when they do, they often fail to re-customize, as their work habits change [8]. Hybrid solutions have been suggested, e.g. [1], where an adaptive mechanism suggests most useful customizations to the user. Meanwhile, adaptation has now been adopted in some mainstream commercial applications. For example, the Start Menu in Microsoft Windows XP™ has an adaptive area that provides automatically generated shortcuts to frequently used applications, thus saving users from having to traverse one or more levels of program menus. Microsoft Office also features Smart Menus – an adaptive mechanism where infrequently used menu items are hidden from view. While no formal study results have been published on either of these interfaces, strong anecdotal evidence exists to suggest that the Start Menu causes few, if any, negative reactions while the Smart Menus in Office inspire strong reactions, both positive and negative, among different users.

OUR ADAPTATIONS

Past work has provided examples of both successful and unsuccessful approaches to adaptation yet very little has been offered in a way of an analysis that would suggest an explanation as to why some of the approaches resulted in improved performance and satisfaction while others were a hindrance to the user. The primary motivation for adapting user interfaces is to improve users’ performance and satisfaction. However, some theoretically beneficial designs, for example the frequency-based adaptive menus [9], proved not to perform well in practice, while theoretically less optimal Split Menus were found to be highly beneficial in one of the studies [11]. Despite that, some projects (e.g. [14]) provide only a theoretical analysis of an adaptation scheme to argue for its adoption.

We hypothesize that users’ subjective perception of the performance of an adaptive UI may be different from the theoretically possible benefits. In particular, we recognize that in addition to a benefit, some users may perceive a cost associated with adaptations: there is a cost associated both with incorrect adaptation and the user needing to become aware of and leverage the adaptation.

We have thus designed three different adaptive techniques that would represent three different points in the cost-benefit space: we expected our Split Interface to have low cost and high benefit, our Moving Interface to have moderate cost and high benefit, while the Visual Popout Interface should have low to moderate cost and low benefit. In order to improve the reliability of our data, we conducted two experiments. The first had realistic and somewhat complex tasks and was designed to elicit realistic subjective responses from the participants. In the second experiment, which was designed to measure the mechanical properties of the adaptation methods, participants were presented with a series of quick repetitive tasks. Further-

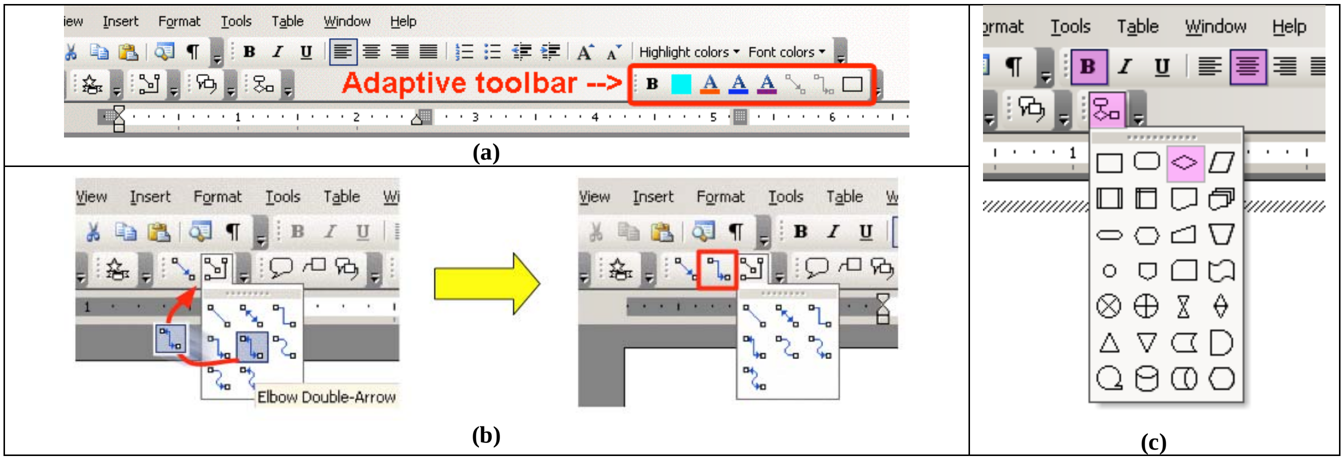

Figure 1. (a) The Split Interface; (b) The Moving Interface; (c) The Visual Popout Interface

more, in the first experiment we subtly varied the predictability of the adaptive algorithm and in the second we varied its accuracy.

Adaptation Types

We built all of our interfaces using .NET automation on top of Microsoft Word XP. This allowed us to explore relatively realistic tasks of varied complexity situated in a moderately complex user interface. In this section, we describe the specific interfaces we tested.

Non-Adaptive Baseline

In our non-adaptive interface we ensured that all the toolbars were wide enough to display all of the buttons at all times and no adaptive behavior was presented to the user.

Split Interface (extra toolbar)

We implemented a version of Gajos' Split Interface [3] for Microsoft Word by including an additional toolbar (Figure 1a). The interface copies important functions onto this toolbar in a spatially stable manner, that is, users could choose either to continue using the (unmodified) original interface or to use the adaptive toolbar. Note that we copied functionality that was originally inside pull-down menu panes as well as the already accessible buttons from the top-level toolbars.

If the adaptive toolbar grows too large (8 buttons in our experiments), functionality is demoted to make space for new promotions. We chose to include a Split Interface in our study because we found no work evaluating the effects of replicating rather than moving content into the extra toolbar. We predicted that this stability should make this interface at least as good as the non-adaptive case.

Moving Interface

Inspired by Shneiderman's concept of moving functionality, our Moving Interface is a variant of our Split Interface. It *moves* promoted functionality from inside popup panes onto the main toolbar, causing the remaining elements in the popup pane to shift and also causing the existing buttons on the toolbar to shift to make space for the promoted button (see Figure 1b). If there are too many buttons already promoted (8 in the first experiment and 4 in the second) on any given row of toolbars, a new promotion will demote some other button, returning it to its original location.

Unlike in our Split Interface, all elements promoted by this adaptation come from inside popup panes thus, from the mechanical point of view, this adaptation offered higher potential benefit to the user than the Split Interface. However, we predicted that the user would perceive Moving Interface as incurring a higher cost due to its spatial instability.

Visual Popout Interface

Our Visual Popout Interface behaves differently still: it highlights promoted buttons in magenta. If a promoted button resides inside a popup menu, both the button invoking the popup menu and the menu item are highlighted as shown in Figure 1c. In our study, no more than 8 buttons may be highlighted at any time.

This interface is related to the baseline interface by Tsandilas et al. [13] and also to Gajos' Altered Prominence UI [3]. We expected it to offer relatively little benefit, while incurring low to moderate costs by changing the appearance of UI elements.

Adaptation Algorithms: Frequency and Predictability

In our *recency-based* algorithm, the N most recently used commands were promoted by the adaptive interface. In our *frequency-based* algorithm, the algorithm computed the most frequently used commands over a short window of interactions (about 20). The latter mechanism resulted in the interfaces adapting a little less frequently (and less predictably) than the former although both adapted in a continuous manner.

EXPERIMENT 1

We set out to compare our three adaptive interfaces to the non-adaptive baseline version within Microsoft Word. We were also interested in exploring the two different adaptation models, but chose to do this between subjects to reduce overall session length.

Participants

Twenty-six volunteers (10 female) aged 25 to 55 (M=46 years) from the Puget Sound community in Washington State participated in this study. All participants had moderate to high experience using computers and were intermediate to expert users of Microsoft Office-style applications, as indicated through a validated screener. Volunteers received software gratuities for participating.

Tasks

We used three tasks of medium complexity, chosen to mimic real-world activities for high external validity. We designed the tasks to be engaging enough that participants did not mind repeating similar versions of them across the different user interface types during a session.

Flowchart

In the flowchart task, participants completed a flowchart of a troubleshooting procedure that was purposefully missing key aspects from its design. Each of these flowcharts, when completed, had 13 components plus connecting arrows. We taught participants how to use the toolbars in Word and provided them with a printout of the completed flowchart they were trying to reproduce. We instructed them to add all of the 16 missing parts but not to spend time aligning the parts precisely as shown on the page. We were interested in their use of the toolbars to add the missing parts, not the exact alignment of the image. We also instructed them to use the cut tool from the top toolbar if they needed to delete anything they might have accidentally added. In order to keep all of the flowchart tasks isomorphic, participants had to add the same number and kinds of elements (3 diamond shapes, 2 rectangles and 11 arrows) to each. No text had to be added.

Quotes

In the quotes task, we presented participants with a table in Word that showed 24 quotes on 4 different topics. Quotes averaged 2 sentences each. We asked participants to edit the table in the following manner: 6 quotes were in the wrong column and participants were to highlight those misplaced quotes with the corresponding color of the correct column header for that topic; the font color of the name of each author needed to be highlighted according to the author's birth date (split into four time periods: before 1 B.C., 1 A.D.-1799, 1800-1899, and 1900-1999).

Country Poster

For this task we presented participants with a one-page, 8.5 × 11 inch, draft poster in Word summarizing information about a country. They had to edit the poster based on their viewing of the completed poster in their instruction packets. In order to maintain task equivalence, all poster tasks required that participants bold, center and enlarge the font of the title (by clicking on the "Enlarge Font" button twice), underline, number, and italicize or bold each of 5 facts, add 4 line callouts to the map from toolbars, add a rounded and a cloud callout to each of the two people in the poster, and drag 6 pieces of text from a sidebar into 6 callouts in the poster.

Equipment

We ran the study on two 3.4 GHz Pentium 4 HP PCs with 1G of RAM to support two simultaneous participants. Each machine drove two NEC MultiSync LCD 1880SX displays set at 1280 × 1024 resolution. Participants used a Compaq keyboard and Microsoft IntelliMouse for input.

Procedure and Design

We ran participants in pairs. At the beginning of the session, we had participants fill out a brief questionnaire about their computer usage experience and habits. We gave them instructions introducing the study and demonstrated each task on a large wall-projected screen. They then performed a set of practice tasks (using the non-adaptive interface), equivalent to those that would be presented in the main part of the experiment. Once they successfully completed the practice task (approximately 10 minutes), we began the study tasks.

The study was a 4 (user interface type: no adaptation, Split, Moving, or Visual Popout) × 2 (adaptation model: frequency or recency-based algorithm) × 3 (task: flowchart, quotes, or country poster) design, with user interface type and task as within-subjects factors and adaptation as a between-subjects factor. During the main part of the experiment, participants performed four isomorphic sets of three tasks, each time with a different interface.

We counterbalanced the presentation order of user interface type and adaptation model using a Latin square design across participants. Participants completed the Flowchart, Quotes, and Poster tasks in the same order – with each of the interfaces that exhibited the same adaptation type but using different underlying adaptation algorithms (i.e., each participant in the pair used a different adaptation model but the same UI look and feel). Between sessions using different interfaces, the experimenter explained how to use the next interface. Since the task sets for each task were isomorphic and the tasks relatively distinct from each other, we expected that interactions across conditions would be more important than task ordering effects. Hence, the order of tasks was kept constant for each user interface condition.

After each of the 4 task series with one user interface, the participants were asked to fill out a brief questionnaire. After the last user interface condition, participants ranked

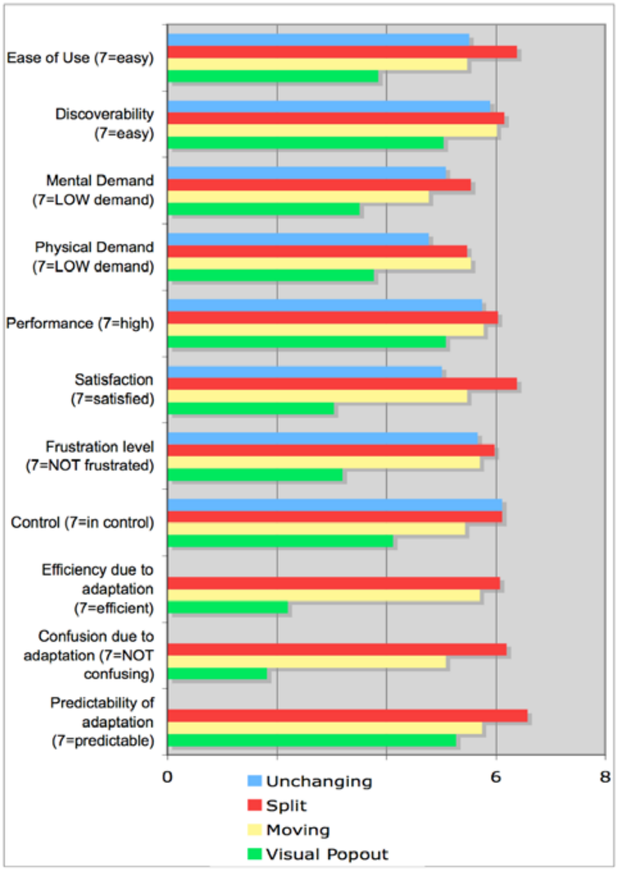

Figure 2. Averages of the responses to the survey questions

the four UI types and explained their first and last choices. Finally, the participants were debriefed. The sessions lasted about 2 hours.

Measures

Dependent variables collected included participant satisfaction ratings, overall preferences, and task times. As expected, the task times were not sensitive enough, and we found no statistical differences using this metric. We discuss results with the other metrics below.

Results

Satisfaction

We used a 4 (UI type: no adpatation, Split, Moving or Visual Popout adaptive UI) × 2 (adaptation model: frequency v. recency) × 11 (questions in the questionnaire) RM-ANOVA to analyze the satisfaction questionnaire ratings. UI type was within-subjects and adaptation model was a between-subjects variable. We found significant main effects for UI type, $F(3,69)=8.5$, $p<.001$ and questionnaire item, $F(10,230)=2.9$, $p=.002$. Post-hoc tests revealed that the main effect for UI type was caused by significantly higher ratings for the Split Interface when compared to either no adaptation or the Visual Popout condition; however there was no significant difference between Split and the Moving adaptive user interfaces. In addition,

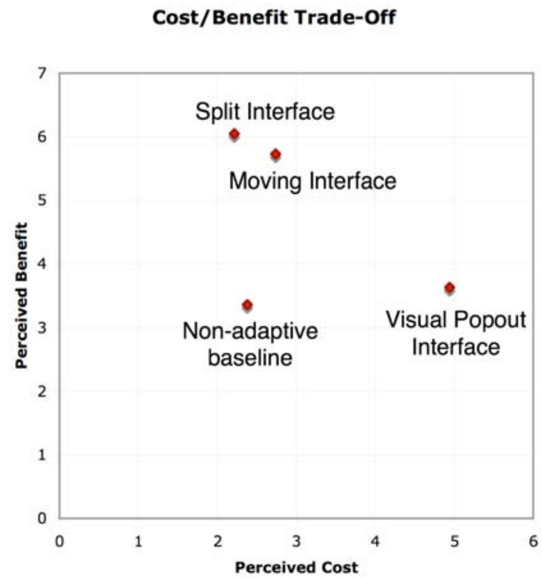

Figure 3. The perceived costs and benefits of different adaptation strategies

the Moving UI was not rated significantly higher than no adaptation. All of the average satisfaction ratings for the user interfaces in the study are shown in Figure 2.

We analyzed the final rankings of the four user interfaces using the Friedman non-parametric test and found a significant preference for the Split UI, $\chi^2(3)=48.4$, $p<.001$.

Perceived Cost and Benefit

We computed an estimate of the participants' perceived costs and benefits associated with the three adaptive interfaces. We used the average of the responses to the Efficiency and Performance questions as a measure of benefit, and the Mental Demand, Physical Demand, Frustration and Confusion questions were used to compute the cost. In the case of cost calculations, we reversed the scale such that 1 would stand for low cost and 7 for high cost. The results are shown in Figure 3 (for the Unchanging Interface, we assigned it a score of 1 for Efficiency due to adaptation and for Confusion due to adaptation). The Split Interface was found most beneficial and least costly despite having lower theoretical benefit than the Moving Interface. The Visual Popout Interface was found to confer little benefit, as expected, but participants found it very distracting and assigned it a higher cost than we had expected.

User Comments

When debriefed, participants confirmed that our tasks did achieve our goal of high external validity, being realistic and engaging. Many participants commented that both Split and Moving interfaces helped them complete the tasks faster. Several participants liked the fact that the

Split interface left the original toolbars unchanged, letting the user decide whether or not to take advantage of the adaptation. A few participants also liked the fact that all functionality related to their current task would end up in one place. Other participants preferred the Moving interface because it put promoted buttons close to their original locations thus letting them discover adaptations opportunistically rather than having to look at the adaptive toolbar. However, some found that same behavior disturbing because it would change the position of other buttons on the toolbars. Not surprisingly, a number of participants also complained about being disoriented when buttons disappeared from popup menus or when the remaining buttons got rearranged. One participant preferred the Visual Popout interface, while others felt that it often made it harder to find what they were looking for.

Summary

This first study was important as it allowed us to observe participants interacting with the user interfaces on tasks that had high external validity. Unsurprisingly, the high variability added by the cognitive decisions made within these tasks rendered task times insensitive to our experimental manipulations. In fact, our task-time analyses did not show any significant differences across the 3 tasks with the exception of the Quotes task, and even then, only the Visual Popout condition was observed to be significantly slower than the other 3 conditions, and this was likely due to implementation issues, hurting performance. Since the satisfaction data showed significant preferences for the split interface condition over no adaptation and highlights, we assumed participants perceived benefits to this kind of adaptation that time measures for these tasks were not sensitive enough to capture. For this reason, we decided to run a second experiment.

EXPERIMENT 2

In Experiment 2, we relaxed our external validity requirements by reducing the cognitive complexity of the tasks. We assumed that while the tasks would be less realistic, having the participant press more buttons while making fewer cognitive decisions would allow us to more carefully measure performance differences that might exist between the user interface types. This also provided us much tighter control over the order of button presses so that we could examine the effectiveness of the adaptation schemes under different predictive accuracy conditions.

Since we found no differences between our frequency and recency-based adaptation models in the Experiment 1, we decided to drop this variable from the study. Additionally, since we could not instrument the Visual Popout interface to be as performant as the other two, we dropped that condition from this study as well.

Participants

Eight researchers (2 female) aged 25 to 58 (M=36 years) participated in this study. All participants had high experience using computers and were expert users of Microsoft Office-style applications. Participants received a small gratuity for participating.

Task and Procedure

At the beginning of each session, the experimenter explained and demonstrated the task and each of the user interfaces. After that the participants worked on a practice task to familiarize themselves with the location of different commands and with the adaptive interfaces. In order to isolate the effects that the various manipulations had on toolbar usage, we used a task in which we told participants exactly which toolbar buttons they had to press (all of those buttons were located inside popup menus accessible from the toolbars). In this task, the system presented an image of a particular command within the Word document. Each participant had to find and hit this button on the toolbar as quickly and accurately as they could. They then hit a "done" button, also presented within the document, and immediately got a new command to target. Each participant repeated this 52 times for each trial, though the first 12 were considered a warm up necessary to initiate (seed) the adaptive models and were not included in the results.

We created two classes of tasks, which resulted in our recency-based adaptive algorithm being either 30% or 70% accurate (i.e., correctly promoting the next button to be clicked by the user) in the case of the two adaptive conditions. This was especially interesting in the Split Interface, where participants could choose either to use the regular toolbar item or the promoted one in the adaptive toolbar. Therefore, the experiment was a 3 (user interface type: no adaptation, Split Interface, or Moving) × 2 (accuracy: 30% or 70%) within-subjects design. All conditions were fully counterbalanced to control for the effects of training.

We ran the experiment on two 2.8 GHz Pentium 4 Compaq PCs with 2G or RAM with the same input/output devices as used in the previous experiment.

Results

Task Times

We ran a 3 (user interface type) × 2 (accuracy: 30% or 70%) × 2 (trial order: 30% first or 70% first) RM-ANOVA, with user interface type and accuracy within-subjects, and trial order run between-subjects.

We found a significant main effect of user interface type, $F(2,12)=8.545$, $p=.005$. Pairwise comparisons using Bonferroni correction revealed that participants were significantly faster using the Split interface, $p=.003$, and marginally faster using the Moving interface, $p=.073$, than without adaptation. The two adaptive interfaces were not significantly different from each other.

We also observed a main effect of accuracy, $F(1,6)=8.859$, $p=.025$. We found a significant interaction between user interface type and accuracy, $F(2,12)=7.689$, $p=.007$, driven by both adaptive interfaces resulting in faster performance (Split: $p<.001$, Moving: $p<.001$) with the 70% accuracy scenario faster than with 30%.

Finally, we saw a significant interaction between accuracy and trial order, $F(1,6)=6.515$ $p=.043$. Participants who saw the 70% condition followed by the 30% condition had a marginal decrease in performance, while participants that saw the 30% first showed vast improvements when they had the 70% condition, $p<.001$. We believe that this is due to various strategies participants built up from using one interface or the other, but verifying this remains future work.

Frequency of Use

Additionally, planned analyses of the Split UI usage data showed that the level of accuracy significantly affected the way participants interacted with the Split Interface, $F(1,6)=10.361$, $p=.018$. On average, when functionality existed in both places, participants utilized functionality from the extra toolbar more frequently in the 70% accuracy condition (M=93.1%) than in the 30% condition (M=81.0%).

Satisfaction and User Comments

After each of the two sections of the experiment, we administered a brief questionnaire, asking the participants how easy each interface made it to find the functionality and how the participant felt it improved his or her efficiency. We used a 3 (user interface type) × 2 (# of questions) RM-ANOVA to analyze the satisfaction questionnaire ratings. We found a significant main effect for UI type, $F(2,30)=4.317$, $p=.023$, explained via post-hoc analyses with Bonferroni corrections by the significant difference between split and unchanging ($p=.035$) and between moving and unchanging ($p=.042$). While participants felt that it was easier to locate functionality in the unchanging interface, they felt that both adaptive interfaces made them more efficient.

In their post-experiment comments, participants focused primarily on three issues: ease of discovery, use of the adapted functionality, and the confusion caused by the adaptive interface. Many found the Split Interface not very useful because it required them to look in two distinct places for any one piece of functionality (unlike in the first experiment, they saw no benefit to having frequently used functionality grouped together). The Moving Interface was considered more convenient in that respect. The Moving Interface, however, caused items in pull-down menus to shift and also caused buttons on the toolbars to move horizontally as functionality was promoted or demoted. Even some of the participants who preferred the Moving Interface overall found this to be a concern.

GENERAL DISCUSSION

The results of our two experiments strongly suggest that purely mechanical properties of an adaptive interface are a poor predictor of a user's performance or satisfaction. Unlike the Split Interface, the Moving Interface promoted only hard-to-reach buttons to the top level so on average it would be expected to save the user the most menu accesses of all the interfaces we evaluated. It did *not*, however, result in a significant improvement in either performance or user satisfaction. In contrast, the Split Interface, which the participants saw as a high benefit and low cost adaptive solution, was strongly preferred over the non-adaptive baseline and resulted in significantly improved performance. We hypothesize that the very low perceived cost of the Split Interface has to do with its high *spatial stability*. That is, this adaptation strategy did not alter the familiar parts of the interface in any way and only adapted a clearly designated separate adaptive area. This design strategy (which differs from the original Split Menus [11] in that items are *copied* rather than *moved* to the adaptive area) is shared by Windows XP Start Menu, the font menus and the symbol chooser in MS Office.

We can begin to consider what other factors influence user acceptance of adaptive UI design. We note that the predictive *accuracy* of an adaptive interface has a significant impact on user performance. As our second experiment demonstrates, the more useful the resulting adaptations, the more likely it is that users will take advantage of the adaptive nature of the interface. This is consistent with previous studies on trust [6,12]. Previous work has also demonstrated that interfaces with higher costs for incorrect adaptation are more sensitive to the differences in the accuracy of the adaptive algorithm [13].

The *frequency of adaptation* appears to have a large impact on the relative weights people assign to the different costs and benefits of adaptation, as illustrated by the conflicting results (both for user satisfaction and performance) of two previous studies of Split Menus ([11] and [2]). The extremely slow pace of adaptation in [11] (once per session) resulted in strongly positive results while the fast pace in [2] (up to once per interaction) caused the same interface to fare worse than the non-adaptive baseline. We suspect that the cause stems from the fact that high frequency effectively reduces a mechanism's *predictability*. We hypothesize that excessively increasing frequency of adaptation will reduce the utility of other adaptive interfaces, but examining this conjecture remains future work.

We also demonstrated that the *frequency of interaction* with the interface and the *cognitive complexity of the task* influence what aspects of the adaptive interface users find relevant. As the differences in user comments between our two experiments suggest, fast and largely mechanical interactions caused users to pay more attention to the operational properties of the interfaces. For example, in the more complex and more slowly-paced interactions of the first experiment, users were less concerned with distance between the extra toolbar and the original location of the adapted buttons but they frequently commented that they appreciated that all relevant functionality was grouped in one place, allowing them concentrate on the task rather than on navigating the interface. This observation should influence how satisfaction data is collected in future studies of AUIs. It also suggests that long-term in situ deployment may result in different user feedback than even a realistic laboratory study and we plan to collect such data for our adaptive interfaces.

Finally, it is surprising that Greenberg's design proved successful, since it drastically restructures the interface after each adaptation. This result might be explained by the very high *complexity of the interface* (a hierarchical menu with over a thousand leaf elements), which prevented the users from developing strong motor memory for the location of different elements.

CONCLUSIONS AND FUTURE WORK

In this paper we reviewed past experiments evaluating different approaches to building adaptive interfaces and observed that the theoretical benefit, due purely to an adaptive interface's mechanical properties, was a poor predictor of the adaptation's success or failure in practice. We observed that in addition to a possible benefit, users perceive adaptations as incurring a cost. Postulating that the balance between perceived cost and benefit would be a better predictor of user acceptance, we designed three adaptive UIs and evaluated them in two experiments. The Split Interface, which was seen as having high benefit and very low cost, resulted in significant improvement in performance and satisfaction over the non-adaptive baseline. The Visual Popout Interface, where the perceived cost exceeded the benefit, was strongly rejected by the participants. Despite promoting only hard-to-access functionality, the Moving Interface was not deemed the most beneficial design.

Through a discussion of our own and past results, we have identified a number of properties of adaptive UIs that are likely to impact the perceived cost and benefits and thus an AUI's acceptance. In particular, this body of work suggests that Split Interfaces, which *duplicate* (rather than *move*) frequently used (but hard to access) functionality to a convenient place tend to improve users' performance and satisfaction, offer medium to high benefits while causing minimal confusion.

Much work remains. We believe that further exploration of the impact of predictability, the frequency of adaptation, and the accuracy of the adaptive mechanism is needed.

ACKNOWLEDGMENTS: We thank Brian Meyers, George Robertson, Georg Petschnigg, and the Visualization and Interaction Group at Microsoft Research. Michael Cafarella, Raphel Hoffmann, Anna Cavender and the anonymous reviewers have provided valuable feedback on the earlier versions of this manuscript. This work was funded in part by a Microsoft Graduate Research Fellowship and an ONR grant N00014-02-1-0932.

REFERENCES

1. Bunt, A., Conati, C., and McGrenere, J. What role can adaptive support play in an adaptable system? In IUI'04, 117-124.
2. Findlater, L., and McGrenere, J. A comparison of static, adaptive, and adaptable menus. In CHI'04.
3. Gajos, K., Christianson, D., Hoffmann, R., Shaked, T., Henning, K., Long, J.J., and Weld, D.S. Fast and robust interface generation for ubiquitous applications. In Ubicomp'05. Springer, 2005.
4. Greenberg, S. and Witten, I.H. Adaptive personalized interfaces: A question of viability. Behaviour and Information Technology. 4(1) pp 31-45. 1985.
5. Jameson, A. and Schwarzkopf, E. Pros and Cons of Controllability: An Empirical Study. In Proc. Adaptive Hypermedia and Adaptive Web-Based Systems. Springer-Verlag, London, 193-202. 2002.
6. Lee, J.D. and K.A. See. Trust in Automation: Designing for Appropriate Reliance. *Human Factors*. 46(1). pp. 50-80. 2004.
7. Mackay, W.E. Patterns of Sharing Customizable Software. In CSCW '90: ACM. 1990.
8. McGrenere, J., Baecker, R. M., and Booth, K. S. An evaluation of a multiple interface design solution for bloated software. In CHI '02. ACM Press.
9. Mitchell, J. and Shneiderman, B. Dynamic versus static menus: an exploratory comparison. *SIGCHI Bull*. 20, 4 (Apr. 1989), 33-37.
10. Oppermann, R. and Simm, H. Adaptability: User-Initiated Individualization. In Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software. R. Oppermann (ed). Lawrence Erlbaum Associates 1994, 54-63
11. Sears, A. and Shneiderman, B. Split menus: effectively using selection frequency to organize menus. ACM Trans. Comput.-Hum. Interact. 1, 1 (Mar. 1994), 27-51.
12. Tiernan, S.L., Cutrell, E., Czerwinski, M., and Hoffman, H. Effective Notification Systems Depend on User Trust. *INTERACT'02*. Tokyo. pp. 684-685.
13. Tsandilas, T. and schraefel, m. c. An empirical assessment of adaptation techniques. In CHI '05 Extended Abstracts. ACM Press. 2009-2012.
14. Warren, J. 2001. Cost/benefit based adaptive dialog: case study using empirical medical practice norms and intelligent split menus. Proc. IEEE AUIC'01, 100-107.