A Field Evaluation of an Adaptable Two-Interface Design for Feature-Rich Software

JOANNA MCGRENERE
University of British Columbia
RONALD M. BAECKER
University of Toronto
and
KELLOGG S. BOOTH
University of British Columbia

Two approaches for supporting personalization in complex software are system-controlled adaptive menus and user-controlled adaptable menus. We evaluate a novel interface design for feature-rich productivity software based on adaptable menus. The design allows the user to easily customize a personalized interface, and also supports quick access to the default interface with all of the standard features. This design was prototyped as a front-end to a commercial word processor. A field experiment investigated users' personalizing behavior and tested the effects of different interface designs on users' satisfaction and their perceived ability to navigate, control, and learn the software. There were two conditions: a commercial word processor with adaptive menus and our prototype with adaptable menus for the same word processor. Our evaluation shows: (1) when provided with a flexible, easy-to-use and easy-to-understand customization mechanism, the majority of users do effectively personalize their interface; and (2) user-controlled interface adaptation with our adaptable menus results in better navigation and learnability, and allows for the adoption of different personalization strategies, as compared to a particular system-controlled adaptive menu system that implements a single strategy. We report qualitative data obtained from interviews and questionnaires with participants in the evaluation in addition to quantitative data.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Evaluation/methodology, interaction styles, theory and methods*

This work is based on an earlier work: An evaluation of a multiple interface design solution for complex software in CHI 2002 ©ACM, 2002, http://doi.acm.org/10.1145/503376.503406.
Authors' addresses: J. McGrenere and K. S. Booth, 2366 Main Mall, Department of Computer Science, University of British Columbia, Canada, V6T 1Z4, email:{joanna; ksbooth}@cs.ubc.ca; R. M. Baecker, 40 St. George Street, Room 7228, Department of Computer Science and Knowledge Media Design Institute, University of Toronto, Ontario, Canada, M5S 2E4, email: rmb@kmdi.toronto.edu.

General Terms: Design, Evaluation, Human Factors

Additional Key Words and Phrases: Human-computer interaction, adaptive interfaces, adaptable interfaces, featurism, bloatware, customization, personalization, individual differences, field experiment.

ACM Reference Format:

McGrenere, J., Baecker, R. M., and Booth, K. S. 2007. A field evaluation of an adaptable two-interface design for feature-rich software. ACM Trans. Comput.-Hum. Interact. 14, 1, Article 3 (May 2007), 43 pages. DOI = 10.1145/1229855.1229858 http://doi.acm.org/10.1145/1229855.1229858

1. INTRODUCTION

Desktop applications such as word processors, spreadsheets, and web browsers have become woven into the daily lives of many people in the developed world. These applications have traditionally started "small" in terms of functionality and have "grown" with each new release. This phenomenon, sometimes called creeping featurism [Hsi and Potts 2000; Norman 1998] or bloatware [Kaufman and Weed 1998], is pervasive: a long feature list is seen as essential for products to compete in the marketplace, applications have become more visually complex, menus have multiplied in size and number, and toolbars have been introduced to reduce complexity but they too have grown in a similar fashion. Insufficient attention has been paid to the impact of this functionality explosion on the user.

We introduce a design that supports two interfaces between which the user can easily toggle: (1) an interface personalized by the user containing desired features only, and (2) the default interface with all of the standard features. The target for the design is feature-rich productivity applications used by a diversity of users. The design has been tested in a prototype front-end to the commercial word processor Microsoft Word 2000 (MSW2K) and evaluated in a field experiment with 20 participants. The two main goals of the evaluation were: (1) to understand the users' personalization behavior with the new design, and (2) to compare our design to the adaptive interface of MSW2K.

Our choice of goals determined our choice of methodology. Our first goal was exploratory in nature; our second goal was comparative. There is a distinction between controlled laboratory evaluation, where statistical significance is the norm, and field evaluation, where qualitative methods are given more weight. We have included comments from participants during interviews, which complement the quantitative data, providing a richer account of their experience during the experiment.

Our evaluation shows: (1) when provided with a flexible, easy-to-use customization mechanism, the majority of users do effectively personalize their interface; and (2) user-controlled interface adaptation with our adaptable menus results in better navigation and learnability, and allows for the adoption of different personalization strategies, as compared to the particular system-controlled adaptive menu system in MSW2K, which implements a single strategy.

1.1 Design Solutions to Complex Software

The traditional "all-in-one" interface has menus and toolbars that are static, so every user, regardless of task or experience, has the same interface. There are a number of alternative interface designs aimed at reducing user interface complexity, although most have received minimal to no evaluation. Design solutions tend to fit into one of two categories: (1) ones that take a level-structured approach [Shneiderman 1997], and (2) ones that offer a personalized interface for each user, most commonly using artificial intelligence.

A classic level-structured design includes two or more interfaces, each containing a predetermined set of functions. The user has the option to select an interface level, but not to select which functions appear in that level. Preliminary research suggests that when an interface is missing even one needed function, the user is forced to the next level of the interface, which results in frustration [McGrenere 2002]. We address this limitation in our design, which uses a level-structured approach while allowing the user to modify the contents of one of the levels. There are a small number of commercial applications that provide a level-structured interface (e.g., Hypercard and Framemaker). Some applications, such as Eudora, provide a level-structured approach across versions by offering both Pro and Lite versions. Such product versioning, however, seems to be motivated more by business considerations than by an attempt to meet user needs.

The Carroll and Carrithers' Training Wheels interface to an early word processor adopts a level-structured-like approach. Although there was only one interface, all of the functionality that was not needed for simple tasks was blocked off such that when the user clicked on a blocked function, a dialog appeared indicating that the function was unavailable in the training wheels system. The design had the user progressing through two distinct phases. After the first phase, the training wheels were removed launching the user into the full system. Novice users were able to accomplish tasks significantly faster and with significantly fewer errors than novice users using the full version [Carroll and Carrithers 1984]. Despite the promise of this early work, mechanisms to support the transition between the blocked and unblocked states were never investigated. In our design, users can move easily back and forth between the designs.

Unlike a classic level-structured user interface, a personalized interface is one that is tailored to each individual user. The two main ways for achieving personalization are through system-initiated adaptation, namely adaptive interfaces, and through user-initiated customization, namely adaptable interfaces. These two approaches have significant differences with respect to the goal of reducing interface complexity. While the broad goal of adaptive and, more generally, intelligent user interfaces is to assist the user by offloading complexity [Miller et al. 1991], a common complaint about adaptive interfaces is that they result in the user perceiving a loss of control [Dieterich et al. 1993; Fischer 1993]. Adaptable interfaces, by contrast, have not typically been designed for the purposes of reducing complexity and so they are often difficult to use. However, they do not suffer the same user control problem [Fischer 1993].

There has been a debate in the user interface design community between those who promote the use of artificial intelligence in the interface and those who promote "comprehensible, predictable, and controllable interfaces that give users the sense of power, mastery, control and accomplishment" [Shneiderman and Maes 1997]. We briefly survey adaptable and adaptive interfaces in turn before describing our design and evaluation.

In terms of adaptable interfaces, many commercial applications allow the user to reconfigure the interface in predetermined ways, such as by adding/removing functions to/from the menus and toolbars, and by moving functions from one menu/toolbar to another. Despite the prevalence of such customization facilities, there has been relatively little research into their design. A common complaint, however, is that the mechanisms for customizing are complex and can therefore require significant time for both learning and doing the customization. Thus, only the more sophisticated users are able to customize. Mackay [1990, 1991] found the latter to be true in the case of UNIX customization. She identified a small group of users, which she called the *translators*, who shared their customizations with the rest of the organization. Others have identified this role and assigned their own names: *tinkerer* [MacLean et al. 1990], and *local developer* [Gantt and Nandi 1992]. By contrast, Page et al. [1996] found that 92% of participants in a large field study customized their word processor. Closer examination of their work shows, however, that a very broad notion of customization was used; for example, changing the zoom setting in a dropdown button on the toolbar was considered a customization. This points to a need for a better understanding of the various forms of customization. In our study, customization is narrowly defined as adding/deleting items to/from the menu/toolbar, which is, at least in many modern graphical user interfaces, significantly more difficult to do than parameter adjustments.

Relative to adaptable interfaces, adaptive interfaces have enjoyed considerable attention by the research community; given the breadth of work, we are unable to do it justice in this short review. Instead, we summarize some relevant trends and highlight selected projects. For greater depth, the reader is referred to Browne et al. [1990] and Schneider-Hufschmidt et al. [1993] for early books on the topic. More recent developments are discussed by Karat et al. [2004].

One well-known limitation of early work in adaptive interfaces was that it was too technology focused–systems were built but relatively little user testing was conducted [Maybury and Wahlster 1999]. This can partially be explained by the fact that evaluation of adaptive interfaces is more complex than that of standard interfaces; there is greater variability with adaptive interfaces and the evaluation methodology needs to accommodate this variability [Greenberg and Witten 1985; Maybury and Wahlster 1999]. Some early examples of adaptive interfaces include the Adaptive Telephone Directory in which the hierarchy of names in the directory adapted to the user's interactions such that the most frequently accessed names were located at the upper levels of the hierarch and the least frequently accessed names were located at the lower levels [Greenberg and Witten 1985]. Adaptive Prompting augmented an interface by providing a permanently visible, dynamic menu that included only the most appropriate and most likely to be chosen actions based on the user's context [Malinowski

et al. 1993]. Both the AIDA system [Cote-Munoz 1993] and the Skill Adaptive Interface [Gong and Salrendy 1995] dynamically adjusted the balance of functionality offered to the user through graphical elements (icons and menus) and the command line, depending on the user's level of expertise. The Eager system detected a repetitive activity and highlighted menus and objects on the screen to indicate what it predicted the user would do next [Cypher 1991].

Of the adaptive designs described above, there was user testing reported for the Adaptive Telephone Directory and Eager. For the former, the results strongly favoured the adaptive directory compared to a static one. User testing of Eager showed promise in its ability to detect and highlight the correct menus and objects, however, "the most striking finding was that all subjects were uncomfortable with giving up control when Eager took over" (p. 37). The work on the Adaptive Telephone Directory is a constrained example, but does provide an existence proof for the efficacy of adaptive interfaces.

Another limitation of early adaptive user interface research is that it focused largely on prototype systems [Thomas and Krogsoeter 1993]. We note some exceptions here. The AID project included an adaptive front end for the British Telecom electronic mail system [Browne et al. 1990]. Among other things, it provided adaptive help based on the user's level of expertise via an application expert. User testing over three half-hour sessions, each separated by three days, showed relatively poor results. An independent expert judged that only 7% of the adaptations made by the system based on inferred user difficulties and expertise were useful. Flexcel was a modified version of MS Excel that provided a separate adaptation toolbar that allowed the user to define new menu entries and new key shortcuts for function parameterization [Krogsoeter et al. 1994]. In addition, there were system-generated adaptation suggestions, which the user accessed at her convenience. User testing showed some acceptance of the adaptation but revealed that the transition between the user accepting system-defined adpation suggestions to actually initiating adaptations him/herself was not satisfactory. Debevc et al.'s [1996] adaptive toolbar for MS Word proposed command icon changes based on frequency and probability of specific command use. User testing comparing the adaptive toolbar to a "fixed toolbar" to which users could somehow add/delete functions showed that the adaptive bar improved performance for certain tasks and that users were generally satisfied with the adaptive bar. (Similar adaptive toolbars for MS Word have been also been proposed [Lim et al. 2005; Miah et al. 1997]). Lastly Linton et al.'s [2000] recommender system alerted users to functionality in MS Word currently being used by co-workers doing similar tasks. No user testing has been reported. We note that all the user testing mentioned above has been done in the lab, and with the exception of the AID project, it has all been single session.

As seen above, the Microsoft Office suite of applications is a common target for adaptive user interface research. It is not surprising therefore that MSW2K introduced an adaptive user interface, namely menus that adapt to each individual's usage [*Microsoft Office 2000 Products Enhancements Guide* 2000]. When a menu is initially opened a "short" menu containing only a subset of the full menu contents is displayed by default. To access the "long" menu one must hover over the menu with the mouse for a few seconds or click on the

arrow icon at the bottom of the short menu. When an item is selected from the long menu, it will then appear in the short menu the next time the menu is invoked. After some period of non-use, menu items will disappear from the short menu but will always be available in the long menu. Users cannot view or change the underlying user model maintained by the system; their only control is to turn the adaptive menus on/off and to reset the data collected in the user model. (Csinger et al. [1994] have investigated the utility of an inspectable user model.)

The work documented in this article aims to address a number of the shortcomings in the literature reported above. We compare an easy-to-use adaptable two-level interface (that was designed specifically to reduce complexity) to the adaptive menus in MSW2K. The adaptable model is a fully functioning front-end to MSW2K that enabled us to conduct a longitudinal field study and collect data reflecting actual personalization behavior as well as self-reported data on preferences.

2. DESIGN OVERVIEW

2.1 Conceptual Design

What makes our design unique is the *combination* of three design elements:

(1) There are two interfaces, one that is personalized (the Personal Interface) and one that contains the full set of functions (the Full Interface); there is a switching mechanism between interfaces that requires only a single button click.
(2) The Personal Interface is *adaptable* by the user with an *easy-to-understand* adaptation mechanism.
(3) The Personal Interface begins small and, unless the user adds many functions, it remains a minimal interface relative to the Full Interface.

The only difference between the two interfaces is the functions that are displayed visually in the menus and toolbars. The set of functions in a particular menu in the Personal Interface is always a subset of those in the same menu in the Full Interface, and the relative ordering of functions is preserved. Thus, the only choice users make with respect to their Personal Interfaces is whether or not to include particular functions.

The conceptual design was proposed by McGrenere and Moore [2000] based on a study of 53 members of the general population who used MSWord 97. They found that while many users were having a negative experience with the feature-richness of the software, the majority of users would not choose a word processor that gave them only the functions that they are currently using. Users want the ability to discover new functions. The proposed design allows users to work in a personalized interface with a reduced feature set while providing one-button access to the standard interface with all features. By default, the Personal Interface is displayed when the application is launched.

There are several reasons for having a user-controlled personalizable interface rather than a predetermined static small interface. Not only do users

typically use very few features [Linton et al. 2000; McGrenere and Moore 2000], but the overlap between the command vocabulary of different users is minimal, even for users in the same group who perform similar tasks and who have similar computer expertise [Greenberg 1993]. The limited command overlap between users suggests that determining appropriate personal interfaces a priori is not possible. Many users do not take advantage of customization features [Mackay 1991], likely because of complexity inherent in customization. This is a primary argument for an *adaptive* interface. Our goal has been to make an easy-to-understand *adaptable* interface instead. By starting users with a small Personal Interface, users are encouraged to customize and take control of their interfaces, right from the outset.

Although our proposed multiple-interface design may seem at first glance to be somewhat awkward and nonintuitive, it was motivated by our earlier research findings and other results from the literature. The experiment was designed to assess the effectiveness of the design and to do a first comparison with the adaptive interface of MSW2K.

2.2 Implementation

Our conceptual design is intended to generalize to any productivity application used by a diversity of users with a broad range of tasks. We chose to implement our design as a front-end to MSWord (1) because word processing tends to be a canonical example in HCI research, (2) because MSWord is relatively easy to program through Visual Basic for Applications (VBA), and (3) because MSWord dominates in the marketplace so we believed that participants would be easy to find.

In order to evaluate this design in a field experiment with participants who were already users of MSW2K, our prototype was implemented so that it did not interfere with any customization that participants had already made to their MSWord interface. It was also designed to be easily installed on top of an existing installation of MSWord. This was accomplished by placing the required VBA code in a specialized document template that was loaded into MSWord on each startup. If necessary, a participant could remove the prototype by simply deleting this template and re-launching Word. The information about function availability in the Personal Interface was stored in a flat file, enabling the prototype to be effectively stateless; this would facilitate the quick reconstruction of a Personal Interface should a problem occur with the software. Figures 1 and 2 show screen captures of the two interfaces as well as the personalizing mechanism.

3. PARTICIPANTS

Twenty experienced MSWord users participated in this evaluation. Participation was solicited electronically through newsgroups and listservs that were broadcast across the University of Toronto campus and the surrounding city. In order to participate, users had to meet the following base criteria: they had to have been using MSW2K for at least 1 month, they had to do the majority of their word processing on a single computer, they had to spend a minimum average of 3 hours word processing per week, they had to have MSWord expertise

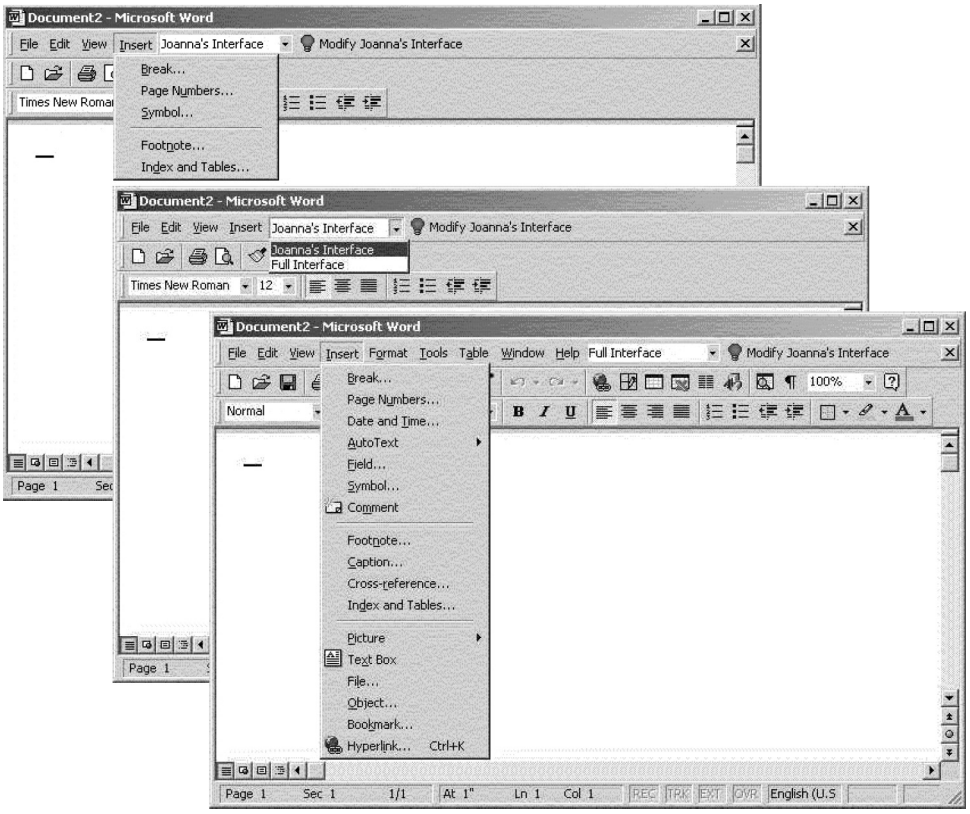

Fig. 1. The two-interface prototype of MSWord. In the first screen shot, the user opens the Insert menu in the Personal Interface. In the second screen, the user invokes the toggle, and will select the Full Interface. In the third screen, the user re-opens the Insert Menu.

above the novice level, they had to be at least 20 years of age, and they had to live at most one half hour's drive from campus. In order to ensure that these criteria were satisfied, prospective participants completed an online screening questionnaire. Ninety-eight people completed this questionnaire. They were considered in the order in which they applied.

A participant's level of expertise was assessed with a Microsoft Office screening questionnaire[1] that was embedded within the online preliminary questionnaire. The screening questionnaire categorizes expertise into five groups: novice, beginner, intermediate, advanced, and expert. A particpant had to be ranked at least as a beginner in order to participate in our evaluation.

Personality differences with respect to feature-rich software were considered. We included 10 *feature-keen* participants and 10 *feature-shy* as assessed by an

[1]The Office Knowledge Test (Version 3) screening questionnaire is a proprietary validated instrument that is used internally at Microsoft for usability evaluations [Davis et al. 1999]. This questionnaire assesses expertise of the whole MSOffice product suite, which includes other applications in addition to MSWord. Although we were particularly interested in MSWord expertise, Microsoft did not have an instrument to assess expertise in MSWord only. This is a limitation of our study.

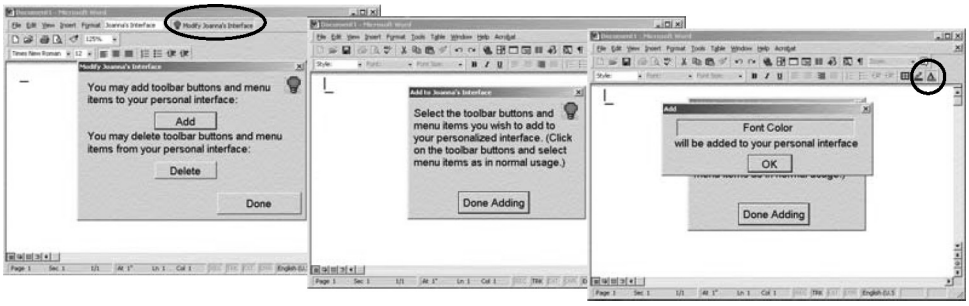

Fig. 2. Process of adding functions to the Personal Interface. In the first screen, the user selects "Modify Joanna's Interface" from the menubar and then "Add" from the dialog box. The user is then prompted to select interface items as in normal usage and is given all elements to choose from. (Those items that are grayed out have already been included in the Personal Interface). The third screen shows that the user has selected the "Font Color" icon.

Table I. Aggregate Description of the 20 Participants

	Gender			Mean	Mean MSOffice	Mean # Years
	M	F	Mean Age (/6)	Education (/7)	Expertise (/5)	Using MSWord
Feature shy	3	7	2.9 = late 20s (1.1 SD)	6.1 (0.9 SD)	3.8 (1.2 SD)	7.1 (4.6 SD)
Feature keen	5	5	2.7 = late 20s (1.0 SD)	5.6 (1.3 SD)	4.5 (1.0 SD)	7.0 (3.2 SD)
TOTAL	8	12	2.8 = late 20s (1.0 SD)	5.9 (1.1 SD)	4.2 (1.1 SD)	7.0 (3.9 SD)

Endpoints on scales are as follows: age 1 = "19 and under", 6 = "60 + "; education 1 = "some high school", 7 = "completed postgraduate degree"; MSOffice expertise 1 = novice, 5 = expert. ($N = 20$).

instrument developed by McGrenere and Moore [2000]. A person is categorized as feature-keen, feature-neutral, or feature-shy based on his/her response to statements about: (1) having many functions in the interface, (2) the desire to have a complete version of MSWord (i.e., not a "Lite" version), and (3) the desire to have an up-to-date version of MSWord.

There was no sampling frame of the user population available to us, so we weren't able to achieve a simple random sample, that is, a representative sample. We have therefore described our sample because it may suggest limits to generalizability. An aggregate description of the participants is found in Table I. The participants are described individually in Table II.

Independent samples t-tests showed that there were no statistically significant differences between the feature-shy and feature-keen participants in terms of gender distribution, age, education, MSOffice expertise, or number of years using MSWord. There were, however, a number of attributes of our sample that lead us to believe that it was not fully representative. On average, the participants appeared to be highly educated (a rating of 5 equals the completion of an undergraduate degree) and long-term users of MSWord. There were no administrative assistants, roles that clearly include many users who do word processing, although an earlier study we conducted did include them. We do not have a definitive reason why we did not get participants in these roles. Perhaps the Call for Participation did not reach these groups or they were reached but individual users did not feel that they could participate in this evaluation. Another likely point of difference between our sample and a representative sample

Table II. Individual Descriptions of the 20 Participants

	Participant #	Occupation	Sex	Age	Education (/7)	MSOffice Expertise (/5)	# Years Using MSWord
Feature shy	1	self employed, administrative computer tasks	F	20–29	5	4	6
	4	software developer	F	20–29	5	3	6
	6	administrator	F	20–29	5	4	4
	7	academic, zoology	M	40–49	7	5	8
	11	interactive architect	F	30–39	6	5	11
	12	interactive architect	F	20–29	6	5	10
	13	self employed, media creation and course development	F	40–49	7	2	3
	14	lawyer	M	50–59	7	5	17
	15	graduate student, psychology	F	20–29	6	2	2
	21	writer	M	30–39	7	3	4
Feature keen	2	technical support and web development	M	20–29	3	4	6
	3	software developer	M	20–29	5	5	6
	5	graduate student, mechanical industrial engineering	F	20–29	6	4	4
	9	graduate student, information studies	M	30–39	6	2	3
	10	graduate student, psychology	F	20–29	6	5	7
	16	academic, information technology management	F	30–39	7	5	14
	17	academic, medicine	M	50–59	7	5	8
	18	self employed, web designer	F	30–39	4	5	10
	20	grad student, computer science	M	20–29	6	5	5
	22	teacher	F	30–39	6	5	8

Participant 8 and Participant 19 were disqualified for not using the prototype during the experiment, which is why they are not shown in the table. (N = 20).

is that graduate students make up one quarter of the participants. This is easily explained by the fact that the Call for Participation was sent to newsgroups on a university campus.

4. EXPERIMENT PROTOCOL

We prioritized our two main evaluation goals as follows:

(1) To understand the participants' personalization behavior with the new design.

(2) To compare our design to the adaptive interface of MSW2K, a commercial interface design for feature-rich software.

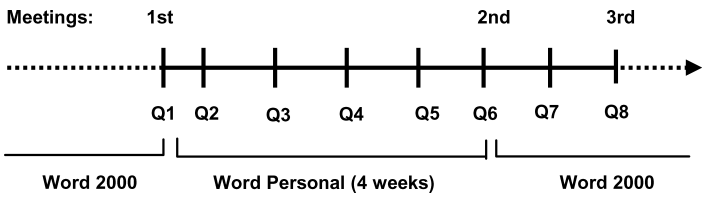

Fig. 3. Timeline of the experiment protocol.

Given this prioritization, a field experiment was chosen instead of a laboratory experiment because it was expected that true personalizing behavior would be significantly more likely to occur with users doing their own tasks in their normal work context rather than in a lab setting with prescribed tasks that would likely be artificial and unfamiliar. To the extent that it was possible, we did introduce some controls in the experiment and so our protocol is best described as a quasi-experimental design.[2]

There are several differences between the two interface designs that we compared, most notably the degree of user control of customization and the use of a dual versus single interface. As will become evident, our study was not designed to tease these factors apart. We instead elected to examine the overall efficacy of an adaptable approach using the combination of factors that we believed offered the best alternative to the existing adaptive interface. Isolating the impact of each factor, once we were able to show that the features together offer an advantage, is an obvious next step.

Each participant was involved for approximately 6 weeks; they used MSW2K prior to the start of the evaluation, worked with our new design for 4 weeks, and returned to MSW2K for the remaining 2 weeks. Participants met with the experimenter on three occasions and completed a series of short on-line questionnaires, Q1–Q8, to assess experience with the software, and a final in-depth semi-structured interview. Figure 3 provides a timeline for the experiment protocol.

First Meeting and Questionnaire Q1. The participant completed a printed version of questionnaire Q1 that assessed the participant's experience with MSW2K. At the same time that the participant was filling in the questionnaire four programs were installed on the participant's machine—the prototype software which we called MSW Personal,[3] a software logger for capturing usage, a small script to transfer the log files to a backup server on the Internet, and a script to delete the prototype in the event of any technical malfunction. Each participant's Personal Interface contained only six functions initially: **Exit** and the list of most-recently-used files in the **File** menu (considered a single function), **Open** and **Save** on the **Standard Toolbar,** and **Font** and **Font Size** on

[2]Quasi-experimental designs are used in natural social settings where full experimental control is lacking [Campbell and Stanley 1972]. This can be contrasted with experimental designs in which there is greater control. The ability to fully control or schedule experimental stimuli—to decide exactly when and to whom stimuli will be applied and the ability to randomize exposures—is what makes a true experiment possible.

[3]In order for participants to consider this prototype as a legitimate word processor we had to give it a seemingly legitimate name in our experiment. MSW Personal was implemented by the first author. It is neither distributed, nor supported by Microsoft Corporation.

the **Formatting Toolbar**. These six functions were selected judgmentally such that two frequently used functions were included for each of the **File** menu and the two toolbars.

Questionnaires Q2 through Q6. These questionnaires assessed MSW Personal. Q2 was completed within 2 days of the First Meeting and was intended to capture the participant's first impression of MSW Personal. Q3, Q4, Q5 and Q6 were completed 1, 2, 3, and 4 weeks respectively from the First Meeting and were intended to capture the participant's experience of MSW Personal over time.

Second Meeting. The Second Meeting was held within 1 day of Q6 being completed. MSW Personal was uninstalled leaving the participant with MSW2K. The log files were collected on diskette.

Questionnaire Q7. Q7 assessed the participant's experience of MSW2K one week following the Second Meeting. It was intended to capture the participant's reaction to returning to MSW2K.

Questionnaire Q8. Q8 asked the participant to rank MSW2K and MSW Personal in terms of each of the dependent measures 2 weeks following the Second Meeting. (In contrast to the first seven questionnaires, which captured participants' feedback on just one of MSW2K or MSW Personal, Q8 captured feedback on both interfaces, in particular rankings of the interfaces.)

Third Meeting. The Third Meeting was held within 1 day of Q8 being completed. The log files were collected on diskette and the participant's machine was completely restored to the state it was in prior to the experiment. A final in-depth semi-structured debriefing interview was conducted with each participant.

Instructions Given to the Participants. In advance of the experiment, participants were only told that some changes would be made to their word processing software but they were not told the nature of the changes. At the First Meeting they were told that a new version of the software had been installed—MSW Personal—which contained two interfaces. The experimenter toggled between the two interfaces once as a brief demonstration. It was pointed out that the Personal Interface contained very few functions initially but that functions could be added or deleted with the Modify button. The process of personalizing, however, was not demonstrated. Participants were told that there was no right or wrong way to use the interfaces and it was specifically mentioned that they could choose to use just one of the two interfaces and essentially ignore the other or they could switch between the interfaces in any way that suited their needs. Participants were not told that MSW Personal would be uninstalled at the Second Meeting. The impression intended was that it would be used for the entire duration of the experiment. In addition to providing the instructions verbally, printouts of the text of the instructions were given. No information about the goals or objectives of the evaluation was provided to the participants at any time during the experiment.

Scheduling. In order to ensure the timely completion of questionnaires and meetings, an individual web page was constructed for each participant that contained all the necessary due dates as well as URLs to all the questionnaires. This acted as a shared resource between the researcher and each participant. In addition, email reminders were sent by 9:00 AM on the due date of each questionnaire with the participant's web page URL directly embedded in the email, which facilitated quick access to the questionnaires. Reminders for each of the three meetings were sent one business day in advance. The participants' web pages were updated regularly to reflect completed activities. There was some flexibility in the scheduling: if a participant was unable to complete a questionnaire or attend a meeting on the scheduled date, the date could be adjusted slightly. Adjustments to the schedule were mostly made during the First Meeting.

Our goal with respect to scheduling was to avoid any confusion about the time and dates of meetings and the due dates of questionnaires. In general, we were very successful in that very few meetings had to be rescheduled during the evaluation and there were few questionnaires that arrived late.

Compensation. Each participant received a $100 gift certificate for a local department store as compensation. In addition, there was one $100 gift certificate awarded as a prize to the participant who completed the most number of questionnaires on time.

Formal Design. The logistical constraints in conducting this experiment in the field precluded the counterbalancing of word processor conditions. The design is a 2 (personality types, between subjects) $\times$ 3 (levels, levels 1,3 = MSW2K, level 2 = MSW Personal, within subjects) design where level 2 is nested with five repetitions.

The fact that there was no control group made this a quasi-experimental design rather than an experimental design [Campbell and Stanley 1972].

5. MEASURES

The dependent measures were based on logging data and data collected from the eight questionnaires. From the logged data, we extracted the total time spent doing word processing, the time spent in each interface, the number of toggles between interfaces, the trace of the modifications made to the Personal Interface, the trace of functions used, and summary statistics of function use. The on-line questionnaires included a number of self-reported measures. Each questionnaire presented the same series of 13 statements that were rated on a five-point Likert scale (Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree). The statements are given below in the order in which they appear in the questionnaires:

[ease of use]	This software is easy to use.
[menu control]	I am in control of the contents of the menus and toolbars.
[learnability]	I will be able to learn how to use all that is offered in this software.
[navigation]	Navigating through the menus and toolbars is easy to do.

[engagement]	This software is engaging.
[match needs]	The contents of the menus and the toolbars match my needs.
[getting started]	Getting started with this version of the software is easy.
[flexibility]	This software is flexible.
[finding options]	Finding the options that I want in the menus and toolbars is easy.
[control]	It is easy to make the software do exactly what I want.
[discoverability]	Discovering new features is easy.
[quickness]	I get my word processing tasks done quickly with this software.
[satisfaction]	This software is satisfying to use.

Q2 through Q6 also included statements specific to MSW Personal. These were rated on the same five-point Likert scale:

[personalizing mechanism easy to use]	The mechanism for adding/deleting functions to/from my Personal Interface is easy to use.
[personalizing mechanism intuitive]	This mechanism for adding/deleting functions to/from my Personal Interface is intuitive.
[personalizing mechanism flexible]	This mechanism for adding/deleting functions to/from my Personal Interface is flexible—I can modify my Personal Interface so it is exactly how I want it.
[toggle easy to use]	This mechanism for switching between interfaces is easy to use.
[concept easy to understand]	This concept of having two interfaces is easy to understand.
[good idea]	Having two interfaces is a good idea.

6. HYPOTHESES

The hypotheses below are categorized according to the two main goals of the evaluation: (1) to understand the participants' personalization behavior with the new design; and (2) to compare our design to the adaptive interface of MSW2K. Pilot testing in the field with 4 participants who each used an earlier version of the prototype for 2–3 months assisted in the formulation of our hypotheses.

6.1 Personalization Behavior with Multiple-Interfaces Design

We wanted to understand whether users could use the MSWord Personal design effectively. Effectiveness relates to being able to personalize according to command usage in a way that is not overly cumbersome. We also wanted to see

whether MSWord Personal is sufficiently flexible to accommodate a variety of personalization strategies.

H1 Usage Hypothesis. The majority of the participants will choose to use their Personal Interface—they will find the personalizing mechanism easy to use, intuitive, and flexible enough such that they will use the mechanism to include all frequently used functions and will spend the majority of their time in their Personal Interface.

H2 Approaches to Personalization Hypothesis. The multiple-interfaces design will allow for different approaches to personalization. Individual differences (feature-shy vs. feature-keen) will influence the strategies adopted.

H3 Growth of Personal Interface Hypothesis. Modifications to participants' Personal Interfaces will be dominated by additions and the size of the Personal Interfaces will reach a steady state. Users will not continually need to modify their personal interfaces.

H4 Modification Triggers Hypothesis. Related to growth, there will be identifiable triggers[4] that prompt participants to modify their Personal Interface. For example, and most obviously, there will be an initial trigger to add functions because the Personal Interface will otherwise be almost unusable.

The last hypothesis, while not directly related to effectiveness and flexibility, was deemed important because if such triggers could be identified, they could provide a design basis for user-assisted personalization.

6.2 Comparison to Adaptive Interface

We hypothesized about how the multiple-interfaces design of MSW Personal compares to one particular instance of an adaptive design, namely MSW2K. While we expected differences in satisfaction, we also expected that because users operate within a small individually constructed interface they would find navigation easier and would feel a greater overall sense of control with MSW Personal. We also expected the ability to learn features would be positively impacted because operating in a user-controlled small interface acts as a training wheels interface to the full interface. With many of these hypotheses, we expected individual differences to play a role.

H5 Satisfaction Hypothesis. Feature-shy participants will be more satisfied with MSW Personal than with MSW2K. The feature-shy will be more satisfied than the feature-keen with MSW Personal.

H6 Navigation Hypothesis. Both feature-shy and feature-keen participants will feel that they are better able to navigate the menus and toolbars with MSW Personal than with MSW2K.

[4]We use the term "trigger" as it was used by Mackay in her study of UNIX customization [Mackay 1991], to mean factors that influence users to modify their interface.

H7 Control Hypothesis. Both feature-shy and feature-keen participants will feel a better sense of control with MSW Personal than with MSW2K.

H8 Learnability Hypothesis. Feature-shy participants will feel that they are better able to learn the available features with MSW Personal than with MSW2K.

H9 Three-way Comparison Hypothesis. When asked to compare their overall preference for MSW Personal, MSW2K, and MSW2K with the adaptive menus turned off (standard all-in-one interface), feature-shy participants will prefer the multiple-interfaces design to an all-in-one design but will prefer all-in-one to adaptive. Feature-keen participants will prefer all-in-one to both the adaptive and multiple-interfaces designs.

7. RESULTS

Most of the logging data for one participant, Participant 9, was lost due to technical reasons. We collected his interface toggling and Personal Interface modification data but missed his function usage data. Where this is relevant, we note $N = 19$; otherwise, $N = 20$ can be assumed unless otherwise stated.

The analysis in Section 7.2, which focuses on uncovering the participants' personalization behavior with MSW Personal, relies on descriptive statistics. The analysis in Section 0, which focuses on the comparison between our design and the adaptive design in Word 2000, includes both inferential and descriptive statistics. Findings that are $p < 0.05$ are deemed to be significant. However, due to the qualitative nature of our field evaluation, our inability to control many of the variables, and our small number of participants, we accept a looser criterion for borderline significant results, using the range of $0.05 \leq p \leq 0.10$. Many qualitative studies are done on so few participants that no analyses can be done; we are able to do some preliminary statistical analyses to suggest phenomena for future investigation.

Both sections conclude with discussions that incorporate the qualitative questionnaire and interview data.

7.1 Experience of Multiple-Interfaces Design

Our primary goal was to understand personalization behavior with the new design. We examine the results for these hypotheses first.

H1 Usage Hypothesis. In the 4 weeks MSWord Personal was installed, the 19 participants each spent on average 596 minutes word processing (SD 554 min, histogram in Figure 4). Fourteen of the participants (74%) spent 50% or more of their word processing time in their Personal Interface (histogram in Figure 5). These same participants added all frequently used functions to their Personal Interface (those functions used on at least half of the usage days).

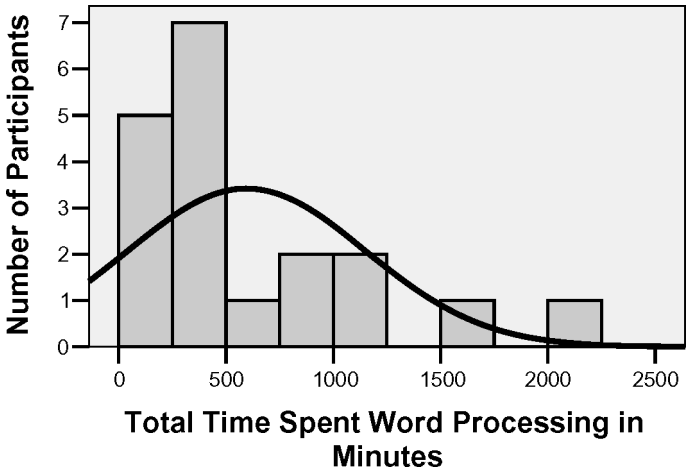

Fig. 4. Histogram of the total time participants spent word processing in 4 weeks with MSWord Personal (N = 19).

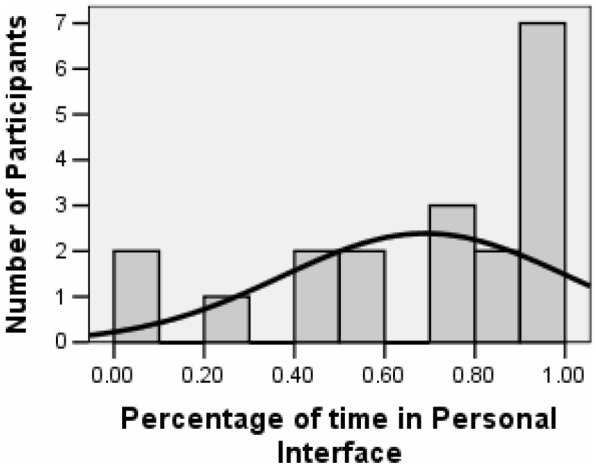

Fig. 5. Histogram of the percentage of time spent in the Personal Interface of the total time spent word processing with MSWord Personal (N = 19).

There are approximately 203 functions in MSW2K that users could include in their Personal Interfaces.[5] Participants did add most of their frequently used functions; the more frequently a function was used, the more likely it was added, as shown in Figure 6. For example, 19 participants had on average 29.8 functions that were used on 25% or fewer of their usage days, and on average participants added 19.7 (66%) of those functions to their Personal Interfaces.

[5]This is an approximate count because some participants had additional MSWord plug-ins loaded, which added functions. In addition, some groups of menu items in VBA are non-standard; although they appear as more than one menu item, they can only be manipulated programmatically as a group. We counted them as a single item. For example, the list of recently used files in the File menu was counted as a single item.

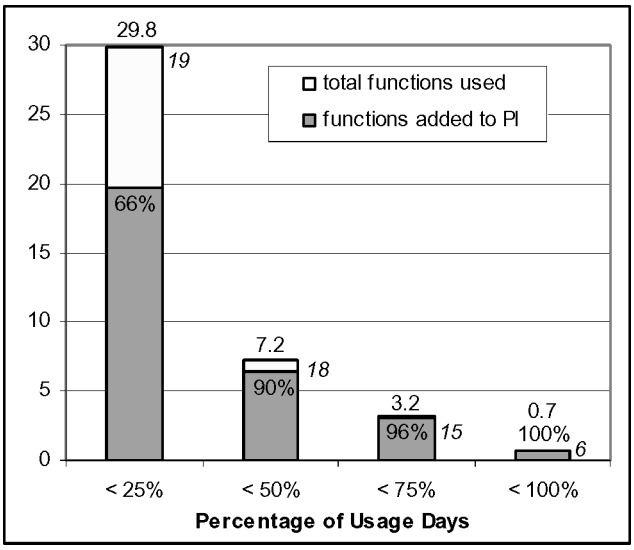

Fig. 6. Mean number of functions used with given frequency. Middle bar indicates the percentage of those functions used that were added to the Personal Interface. Number in italics gives the number of participants who had at least one function used at that frequency (N = 19).

The percentage of functions added increases to 90%, 96%, and 100% respectively for the next three quartiles.

Although the majority of the frequently used functions were added to the Personal Interfaces, there were relatively few such functions. Figure 6 shows that on average participants only had 0.7 functions that were used 75% or more of their usage days.[6]

Questionnaire data indicated that participants found the personalizing mechanism easy to use, intuitive, and flexible. These three attributes had mean ratings out of 5 of 4.3, 4.1, and 4.0, respectively.

Note that all the data reported above represents aggregate data for both the feature-keen and the feature-shy participants. Independent sample t-tests were first run to see if there were any significant differences between the groups of participants for the dependent measures investigated, and no statistically significant differences were revealed.

The largest Personal Interface was 75 functions, by a feature-keen participant. There was no correlation between time spent using the MSWord Personal and the number of functions added to the Personal Interface.

[6]At first glance, these numbers may appear to be too low to be correct; after all, it should be safe to assume that all users must at least have to use Save and Close every day. However, recall that only menu items and toolbar items were counted, that is, those items that users can personalize with our system. Users often use the "X" button in the upper right-hand corner to close a window and use the hotkey "Ctrl-S" to save a document. Because these common shortcuts are not included in a menu or toolbar, they are not included in these counts. Hotkeys are not menu items that come and go in adaptive menus; they are always available. They were kept always available in both conditions in our experiment to ensure that participants would be able to use MSWord in both conditions much as they normally would.

Table III. Desired Approach to Personalization

		Feature Shy	Feature Keen	Total
Approach to Personalization	frequently-used, upfront	2	–	2
	frequently-used, as-you-go	1	3	4
	all, upfront	–	4	4
	all, as-you-go	3	–	3
	frequently-used, as-you-go, gave up	1	–	1
	all, as-you-go, gave up	2	3	5
	none, gave up	1	–	1
	Total	10	10	20
Summary				
Which functions?	Frequently-used	4	3	7
	all	5	7	12
	none	1	–	1
	Total	10	10	20
Added when?	Upfront	2	4	6
	as-you-go	7	6	13
	none	1	–	1
	Total	10	10	20
Gave up?	Gave up	4	3	7

Terminology as follows: **frequently-used** = add frequently-used functions only; **all** = add all functions that are used; **upfront** = add majority of functions right away; **as-you-go** = add functions as they are needed; **none** = participant did not personalize; **gave up** = abandoned desired approach. (N = 20).

Summary: Participants personalized their interfaces according to the frequency with which they used functions; 74% of the participants spent 50% or more of their time in their Personal Interface; and participants agreed that the personalizing mechanism was easy to use, intuitive and flexible.

Hypothesis Supported: Yes.

H2 Approach to Using Two Interfaces. Participants were not told how they should use the two interfaces in MSW Personal and were therefore able to approach it in any way that met their needs. Analysis of the log data and the debriefing interviews allows us to approximately discern the general approaches that participants took to constructing their Personal Interfaces. In general, each approach can be broken down into two independent components: (1) *which* functions were added, namely, only the most *frequently-used* functions or *all* used functions, and (2) *when* functions were added, namely, *upfront* within the first few days of usage or gradually as functions were needed (*as-you-go*). The top of Table III gives a detailed breakdown, and an overall summary is given at the bottom of the table.

We look first at *which* functions were added. Including both participant groups, we see that participants were almost twice as likely to add all used functions (12 participants) as only the frequently used functions (7 participants). Participants who added all used functions generally expected to use their Personal Interface exclusively whereas those who added only regularly used functions expected to switch to the Full Interface when irregularly used functions were needed. Taking individual differences into account, feature-shy were unexpectedly almost evenly divided between only including frequently

used functions (4 participants) and all functions (5 participants). We thought feature-shy participants would want as minimal an interface as possible; that is, to only add their frequently used functions. Feature-keen participants, on the other hand, were more than twice as likely to want all their used functions in their Personal Interface (7 participants) rather than just the ones they used frequently (3 participants). Their preference for all functions was expected.

In terms of *when*, the functions were added, including both participant groups, 6 participants took the approach of adding the great majority of functions that they expected to add upfront, and then only rarely adding additional functions. By contrast, 13 participants took the approach of adding some functions in the beginning and then gradually adding additional functions (as-you-go). Accounting for individual differences, feature-shy participants clearly favored the add as-you-go strategy (7 participants) to the add upfront strategy (2 participants). Feature-keen participants also appeared to favor the add as-you-go strategy (6 participants to 4) but not as decisively as the feature-shy.

Seven participants *gave up* on their desired approach to some degree. For most this meant that they stopped personalizing, or only added very few functions beyond their first few days of using MSW Personal. These participants used their Personal Interface to the extent that they could but would then switch to the Full Interface when a function not available in their Personal Interface was needed rather than taking the time to add it. For example, Participant 1 is categorized as "all, as-you-go, gave up." She wanted to have all the functions she used in her Personal Interface but in the end she realized she was using a lot more functions than she expected, some of which she was learning for a new contract she started during the evaluation. She did continue to personalize throughout the experiment but ended up just adding the most frequently used functions, which was not her desired approach. More typical behavior of participants who gave up is described here by a participant who is categorized as "all, as-you-go, gave up":

I would start out of the personal. At the beginning I was adding things pretty regularly to the personal but I felt that I was just continually adding things and so eventually I would just start out and use the personal as long as it was convenient and then I would just switch once I felt like I needed to add another function. [Interview, Participant 10]

Participant 14 was categorized as "none, gave up" because he used the Full Interface almost exclusively and clearly wasn't willing to spend the time to explore his Personal Interface. It was obvious from his comments in the debriefing interview that he really did not understand the concept of a Personal Interface and by the time of the interview he had completely forgotten that he himself had added four functions to his Personal Interface during the First Meeting.

Summary: Participants adopted various approaches to personalization in terms of which functions were added to their Personal Interfaces and when they were added. No one strategy dominated outright. *All*, *as-you-go* was the most popular (8 participants), but the other three combinations were also adopted: *frequently-used*, *upfront* (2 participants); *frequently-used*, *as-you-go*

Table IV. Aggregate Personalizing Data (N = 12)

	Minimum	Maximum	Mean	SD
Last day of initial period	1	5	2.75	1.22
Percentage of functions added by end of initial period	44.00	100.00	81.58	19.30
Number of days that participant personalized	3	6	3.83	1.11
Total number of days MSW Personal was used	6	26	16.50	5.25
Day at which 90% of functions have been added	1	13	4.75	3.70
Percentage of 4 weeks at which 90% have been added	5.00	87.00	31.32	26.21
Percentage of 4 weeks at which last personalization done	15.00	100.00	74.65	24.75

(5 participants); and *all*, *upfront* (4 participants). Individual differences appear to play a role in the strategy adopted, but not a decisive role. Feature-keen participants were more likely to add all used functions, rather than just their frequently used functions. Seven participants gave up to some extent on their preferred approach.

Hypothesis Supported: Partially. The multiple-interfaces design did allow for different approaches to personalization; but individual differences did not strongly influence strategy.

H3 Growth of Personal Interface Hypothesis. The size of participants' Personal Interfaces, with the exception of seven deleted functions, increased monotonically. All participants added functions and only 3 participants deleted a combined total of seven functions.

For the purposes of understanding how Personal Interfaces evolved, we directed our attention exclusively to the 13 participants who did not give up their desired approach to personalization as these participants set up their Personal Interface in a way that met their needs. This group includes Participant 9 for whom we only have partial logging data and so we omit his data from this analysis, which leaves 12 participants. For this group of participants, there was an initial period when modifications were made regularly. This series can be defined by a first modification followed by subsequent modifications that were at most 2 usage days from the previous modification. Table IV shows that the average duration of this initial period lasted 2.8 days, and for the participant who had the longest initial period it only lasted 5 days. Of the total number of functions that these participants added during the 4 weeks, on average each had added 82% of his/her total by the end of the initial period.

On average, participants personalized on 3.8 of the days that word processing occurred, with the participant who most frequently personalized doing so on 6 days. The size of participants' Personal Interfaces did approximately reach steady state in that there was a point at which the size of the Personal Interfaces did not increase/decrease by more than 10%. We expected this point to be within the initial period for the majority of the 12 participants. The results show, however, that the steady state point was reached by the end of the initial period for only half of the 12 participants (for 2 participants steady state was equal to

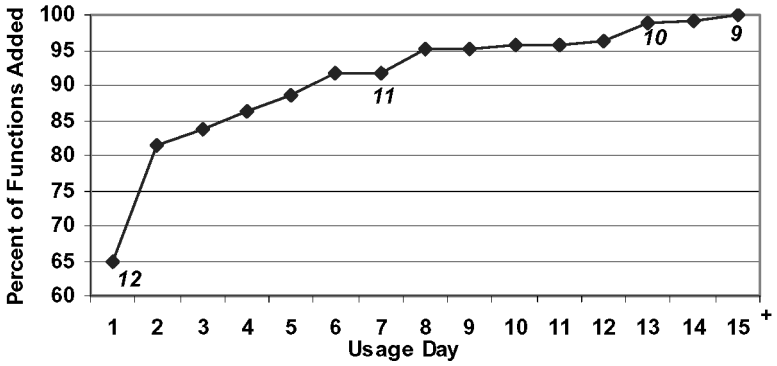

Fig. 7. Percentage of functions added by a given usage day, where 100% equals 485 functions. Note that the y-axis begins at 60 percent. The numbers given inside the plot area indicate how many of the 12 participants used MSW Personal for a given usage day. For example, the 11 indicates that only 11 of the participants had 7 or more usage days. (N = 12).

the initial period and for 4 it was less than the initial period). On average this steady state period came after 4.8 days of usage, which was on average 31% of the way through the 4 weeks in which MSW Personal was used. On average, the last modification came 75% through the 4 weeks.

The data can be viewed on a day-by-day basis. The cumulative total number of functions added by the 12 participants was 485. Figure 7 shows that within the first 2 days that MSW Personal was used, 81% of all 485 functions had been added.

Summary: Personalization was dominated by additions; participants added on average 90% or more of the functions that they were going to add within 4.8 usage days, but participants were on average not finished personalizing until almost three quarters of the way through the 4 weeks. Participants were not continually personalizing.

Hypothesis Supported: Yes.

H4 Modification Triggers Hypothesis. In order to evaluate this hypothesis we began by identifying the patterns of usage with respect to addition, in particular, whether a function was used before or after it was added and, if so, how soon it was used before or after if was added. A detailed analysis of these usage patterns enabled us to define four triggers for addition:

Immediate-need trigger. The user has an immediate need to use a function that is not currently in his/her Personal Interface and therefore adds it. This is shown clearly in the log file by the pattern: <add function x> followed directly by <use function x>.

Initial trigger. The user desires to add functions when first using MSW Personal. Any function added within the first 2 days of usage that does not satisfy the immediate-need-trigger satisfies this trigger.

Previously-used trigger. The user has already used a function and expects to use it in the future. Any function that has been used before it is added

Table V. Triggers for Adding Functions to the Personal Interface (N = 485)

Triggers	Number of Functions that Satisfy Trigger	% of Total
Immediate-need	43	8.87
Initial	372	76.70
Previously-used	6	1.24
Future-use	64	13.20
TOTAL	485	100.00

and does not satisfy the immediate-need or initial triggers satisfies this trigger.

Future-use trigger. The user expects to use a function in the future and so adds it to the Personal Interface. Any function that does not satisfy any of the first three triggers satisfies this trigger.

Table V summarizes our findings with respect to triggers for the addition of functions. We see that the initial trigger and the immediate-need trigger together accounted for almost 86% of the functions added. The previously used and future-use triggers accounted for the addition of the remaining functions, which were typically added while an immediately-needed function was being added.

There were two triggers for deletion:

Mistaken-addition trigger. A function was added by mistake. It was not intended or was the wrong function.

Non-use trigger. A function is not being used.

Only 3 participants deleted a combined total of seven functions: one, two and four functions respectively. In all seven cases the function was deleted directly after it was added. When queried in the debriefing interview the participant who had deleted four functions indicated that she had been testing the personalizing mechanism, and the other 2 participants said that the deleted functions had initially been added by mistake. Thus, the mistaken-addition trigger applied to the deletion of three functions and the non-use trigger did not apply to any deleted functions. One might initially assume that the lack of the non-use trigger is explainable by the fact that MSW Personal was only used for one month. To counter this, however, the great majority of participants indicated in the debriefing interview that they would not likely have bothered to delete functions even if they were not being used.

Finally, to double check that our participants were not simply customizing when reminded of their study participation, we investigated a potential correlation between the times when subjects personalized their interfaces and times when questionnaires were due. For questionnaires Q1 and Q2, 68% and 63% of the subjects respectively did perform at least one customization on the days when those questionnaires were answered. This is not surprising, however, given the minimal starting Personal Interface and the strong initial trigger. Excluding customization that overlapped with Q1 and Q2, there was only an average 17.3% overlap of customization with Q3 through Q6 ($N = 19$). If we consider only those who didn't give up, that average is only slightly higher at

20.1% ($N = 12$). Thus, the data suggests that personalization was not being triggered by questionnaire completion but was done at various times during the roughly six days preceding the days when each of the Q3, Q4, Q5 and Q6 questionnaires were answered.

Summary: The initial trigger accounted for the majority of additions (77%). When participants added an immediately needed function (9%), they would typically also add a function they expected to use in the future (13%) or one they had already used (1%). Seven functions were deleted, four to test the mechanism and three because they had originally been added by mistake.

Hypothesis Supported: Yes.

Discussion and Additional Qualitative Feedback. Here we discuss our findings about personalization behavior, and specifically how the multiple-interfaces design impacted this behavior. We include selected comments made by the participants about MSW Personal, both from the open-ended sections in questionnaires Q1 through Q8 and in the final debriefing interview. The goal is to bring the quantitative data to life by placing it in the context of the qualitative data.

Approach to Personalization. MSWord Personal appears to provide sufficient flexibility to allow users to personalize as they see fit. Of the 13 participants who did not give up, there was almost an even distribution between the four combinations of *when* and *which*: frequently-used, upfront (2); frequently-used, as-you-go (4); all, upfront (4); all, as-you-go (3). We speculate that this flexibility played a key role in the success of the multiple-interfaces design.

Having said that, it is interesting to note that none of the participants who took the approach of adding functions upfront ended up giving up. This suggests that personalizing upfront may be a more effective strategy than as-you-go.

User-Assisted Personalization. Four participants suggested having some automated assistance in the construction of their Personal Interfaces, but some of them did acknowledge loss of control as a potential negative side effect. Interestingly, the quote below is from Participant 7; he was annoyed by the adaptive menus and yet he was still hopeful that the Personal Interface could be built automatically.

So if I could have something where it automatically creates the personal based on my use without having to point and click buttons around, which is easy enough but a bit of a pain, if it automatically could do it for me so that over time I created a personal interface by default so my usage pattern creates it without that annoying short menu stuff, that would be nice. Because then I wouldn't actually have to think about it, I'd just use Word and it would create it as I go. [Interview, Participant 7]

It is not yet clear whether fully automated interface construction is effective, but user-assisted personalization has potential. Knowledge about customization triggers should play a role in how the assistance is designed. We return to this topic in the final section of this paper.

Role of the Full Interface. We checked whether having the Full Interface was considered important. Customizable interfaces do not generally provide easy

access to the entire set of functionality, except through a full reset, which results in loss of the customization effort. We found that having the Full Interface was generally well liked. While there were two participants who did not make use of it at all, the great majority did use it at least once and appreciated having it available. Example representative comments include:

I'd always want to have the full interface to go to just as like a baseline kind of thing. . . .Because that Word [the full interface] was there it was this safety net of—yah I know it's over there if I ever do need it anyways. [Interview, Participant 11]

I like the security blanket of having the full interface but over a longer period of time I probably would have extinguished my use of it, pretty much. [Interview, Participant 17]

What I would really hate is if the personal one—if you couldn't go back and forth. The fact that you could back and forth and that it was so easy to go back and forth, that was very good. [Interview, Participant 13]

So although our evaluation did not assess an easy-to-customize Personal Interface in the absence of an easy-to-access Full Interface, our data suggests that the Full Interface is an integral part of the design and that personalizing behavior would be significantly altered without having a full interface available.

Initial Personal Interface. We initialized the Personal Interface to be small with the goal of forcing customization. Two participants felt that the Personal Interface should have initially included more functions—those functions that are used by everyone. For example:

If there was an interface, maybe it differs by industry, I don't know, but with the most common functions—like print. Everyone prints. Everyone makes things bold. So if there was a kind of pre-selected simple menu that wasn't overwhelming. I would maybe be tempted to prefer something like that. Like it would make my decision harder to decide should I use the full 2000 interface or the personal interface and then be able to switch. That distinction would be a little less clear in my mind. So if there was a pre-selected bunch of functions that everyone happens to use and I happen to fit into that everyone category. . . [Interview, Participant 3]

The implicit assumption by these participants is that such a set exists. Research by Greenberg into UNIX command usage showed, however, that there is only minimal command overlap even between participants within the same group who are performing similar tasks [Greenberg 1993]. We expect that similar results would be found for MSWord command usage,[7] but additional research would be required to substantiate that claim.

Assuming for the moment the existence of a "reasonable" set of functions that could be used to initialize the Personal Interface, it would be interesting to see

[7]Linton et al. [2000] investigated command usage in MSWord 6.0, however they did not analyze the data on the basis of individual subjects, but rather aggregated it across all subjects.

how personalizing behavior might change with such a relatively large initial Personal Interface. On the one hand, users would not have to take the time to do any initial customization. On the other hand, users would not be taking ownership of their Personal Interfaces from the outset, which could diminish their overall engagement in the personalization process. Another possible research avenue would be to investigate appropriate initial Personal Interfaces based on individual differences (feature-keen/shy).

Usability of MSWord Personal. Our participants identified some basic usability improvements to the implementation of the personalizing mechanism.

Three participants commented that it was somewhat cumbersome. The confirming dialog box that appears after the selection of each function was seen to be unnecessary. For example:

The double menu that you get... I found confusing, a little bit, but it was easy to use. ... It's a little clunky. [Interview, Participant 17]

I mean it wasn't difficult. It was just you had to click, and you had to click again, and click again, and go back and go forth—it was just bulky. [Interview, Participant 22]

Three participants wanted to be able to add an entire menu at once. The current design requires each menu item to be added one at a time, and because of the number of steps involved this can be time consuming if the majority of a menu is desired.

Four participants felt that MSW Personal was "a good start" but in addition to simply selecting a subset of functions for their Personal Interfaces, they wanted to be able to restructure the menu hierarchies:

I would like to be able to rewrite the stupid menu structure of the MS Word program, not just select the options that I want within the stupid tree structure. [Q4, Participant 7]

I would have liked to put things in different places you know. And this is bad because I do this in Word because I don't like what they have decided is on this menu. [Interview, Participant 11]

Both of these participants were surprised when they were informed at the end of the evaluation that this restructuring functionality was available through MSWord's native customization facility. (It is worth noting that these comments were made by Participants 7 and 11, both of whom are expert long-term users of MSWord.)

One participant requested the ability to have more than two interfaces—she wanted different Personal Interfaces related to the different tasks she performed:

One thing that would have been cool is if I could have had different settings. Like if you have the default Word and then a personal... because I work on charts and I work on just regular reports. If I could have two kind of different settings—say this is going to be my flowchart settings. And those would have draw, all the draw and shape tools and everything. And then this is my report interface and

that would just have regular stuff. That would have been cool. And I don't know if a lot of people work like that. Because sometimes . . . I'm just writing reports or writing proposals and stuff like that and then other times I'm just doing very different things and I need to switch my orientation around and like have all this drawing stuff. And that just doesn't fit with regular writing. [Interview, Participant 11][8]

Our design goal for the customization mechanism was to make it straightforward/understandable so we opted for a design that offered only basic functionality (adding/deleting functions) and that could be learned quickly through trial and error. Despite the comments above, all participants noted how easy it was to use the addition/deletion mechanism. Thus, we believe that it was easy to use in the sense that it was easy to figure out what to do and no errors occurred, but there were too many steps required. One participant pinpointed our trade-off: "The Add/Delete procedure seems slow and redundant for some reason, but is rather idiot-proof." We could rectify the "clunkiness" by removing the confirmation dialog box and designing a new form of menu such that when the user is selecting items from a menu to add to the Personal Interface, the menu stays open and check boxes appear adjacent to each item indicating its availability in the Personal Interface. Currently, in order to add a menu item the user selects the item as in normal menu usage; after selection, the menu disappears and the user must reopen the menu in order to add another item.

We believe that personalization was affected by the design of the personalizing mechanism. Some participants would not likely have given up if functions could have been added more quickly. Some wanted more flexibility, but to support menu hierarchy restructuring and more than one Personal Interface would likely make the personalizing mechanism inaccessible to nonadvanced users. The native MSWord customization facility does allow for some of this, but relative to our personalizing mechanism it requires substantially more skill to use. We had expected the flexibility ratings recorded in the questionnaires to reflect the limitations of the personalizing mechanism, but the participants rated flexibility 4.0 out of 5.0. So although some users did articulate preferences for additional flexibility, the quantitative data shows that the mechanism was sufficiently flexible for most participants. One alternative design to explore would be two levels of customization: basic and advanced. Greater flexibility would be available through the advanced level, but users would by default start in the basic level.

Individual Differences. The expected differences between the feature-shy and the feature-keen participants did not play out in any substantial way in how they personalized and used MSW Personal and what they had to say about their experience using it. Significant differences between the two groups of participants did appear, however, in terms of how MSW Personal was compared to other interfaces, as will be shown in the next section.

[8]We have considered other "bases for personalization" in addition to reduced functionality sets. These include task-based personalization and the use of digital personas. See McGrenere and Moore [2000].

7.2 Comparison with the Adaptive Interface

We turn now to the remaining hypotheses, which cover our secondary evaluation goal, namely to compare the multiple-interfaces design of MSW Personal to the adaptive interface of MSW2K.

The first four of these hypotheses (H5–H8) compare the two interfaces with respect to satisfaction, navigation, control, and learning. The means of each of these four dependent measures at Q1 through Q7, separated by personality type, are given in Figure 8. A series of three factorial ANOVAs (Analysis of Variance) was run to test for significant differences:

(1) Q1 vs. Q6: Compares measures after extended time in each condition. Q1 responses reflect usage of 1 month or more with MSW2K. Q6 reflects 1 month's use of MSW Personal.
(2) Q6 vs. Q7: Compares measures as an initial reaction of returning to MSW2K after 1 month's use of MSW Personal.
(3) Q2, Q3, Q4, Q5, Q6: Compares measures at regular intervals during 4-week usage of MSW Personal.

In addition to reporting statistical significance, we report the effect size eta-squared (η^2), which is a measure of the magnitude of the effect of a difference that is independent of sample size. Both Landauer [1997] and Vicente and Torenvliet [2000] note that effect size is often more appropriate than statistical significance in applied research such as Human-Computer Interaction. The commonly accepted metric for interpreting eta-squared is: 0.01 is a small effect, 0.06 is medium, and 0.14 is large.

H5 Satisfaction Hypothesis. The MSWord versions impacted the satisfaction of the two groups of participants differently (Figure 8). There was a borderline significant cross-over interaction for Q1 vs. Q6 ($F(1,18) = 4.12$, $MSE = 0.98$, $p = 0.057$, $\eta^2 = 0.19$) prompting us to test the simple effects for each group of participants independently. The Q1 vs. Q6 comparison was not significant for the feature-keen participants, however, the increase in satisfaction was borderline significant for the feature-shy ($F(1,9) = 3.65$, $MSE = 1.34$, $p = 0.089$, $\eta^2 = 0.29$). Two further tests compared the satisfaction of the feature-shy participants to the feature-keen participants at Q1 and then at Q6. The feature-shy were found to be (borderline) significantly less satisfied than the feature-keen while using MSW2K at Q1 ($t(18) = -2.04$, $p = 0.056$). However, there was no significant difference detected between the two groups while using MSW Personal at Q6.

When participants returned to MSW2K (Q6 to Q7), the feature-shy appear to have dropped in satisfaction and the feature-keen had effectively no change, but this cross-over interaction was not significant.

Summary: The analysis suggests that the feature-shy participants were less satisfied than the feature-keen participants when using MSW2K, however, the feature-shy participants experienced an increase in satisfaction while using MSW Personal. The feature-keen participants did not experience any change in satisfaction when they switched to MSW Personal.

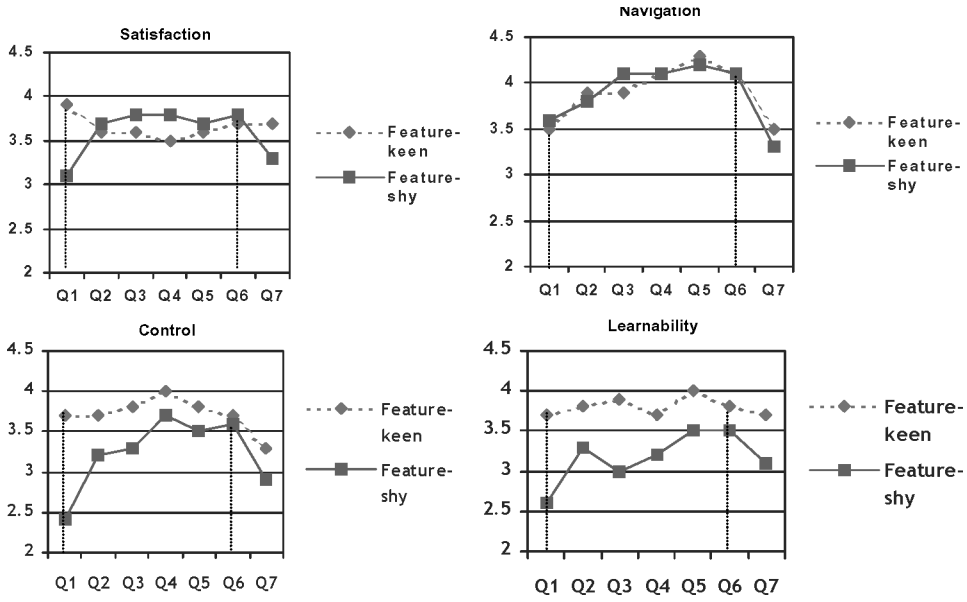

Fig. 8. Mean values of satisfaction, navigation, control, and learnability, separated by personality type, for Q1 through Q7. Grey bars: Q1 reflects responses about MSW2K at outset of the evaluation; Q6 reflects responses after one month's use of MSW Personal. (N = 20).

Hypothesis Supported: Partially. Feature-shy participants were more satisfied with MSW Personal than with MSW2K, but they were not more satisfied with MSWord Personal than the feature-keen participants.

H6 Navigation Hypothesis. The version of MSWord had a significant main effect on participants' perceived ability to navigate in both the Q1 vs. Q6 comparison ($F(1,18) = 5.76$, $MSE = 1.05$, $p = 0.027$, $\eta^2 = 0.24$) and the Q6 vs. Q7 comparison ($F(1,18) = 8.02$, $MSE = 1.22$, $p = 0.011$, $\eta^2 = 0.31$) (Figure 8). Both comparisons favored MSW Personal. There was a borderline significant learning effect in Q2 through Q6 ($F(4,72) = 2.38$, $MSE = 0.18$, $p = 0.06$, $\eta^2 = 0.12$) indicating that navigation became easier over time; unsurprisingly, none of the post hoc pairwise comparisons with the Bonferonni error correction were significant.

Summary: The analysis suggests that both the feature-keen and the feature-shy participants found it easier to navigate the menus and the toolbars using MSW Personal than MSW2K.

Hypothesis Supported: Yes.

H7 Control Hypothesis. The results of the Q1 vs. Q6 comparison of control are dominated by a borderline significant interaction ($F(1,18) = 4.38$, $MSE = 0.82$, $p = 0.051$, $\eta^2 = 0.20$) (Figure 8). Testing the simple effects found the Q1 vs. Q6 comparison to be nonsignificant for the feature-keen participants, however, the feature-shy perceived a significant increase in control ($F(1,9) = 11.17$, $MSE = 0.64$, $p = 0.009$, $\eta^2 = 0.55$). Two further tests compared control for the feature-shy participants to the feature-keen participants at Q1 and then

at Q6. The feature-shy reported significantly less control than the feature-keen while using MSW2K at Q1 ($t(18) = -2.72$, p = 0.014). However, there was no significant difference detected between the two groups while using MSW Personal at Q6.

There was a main effect for control from Q6 to Q7 ($F(1,18) = 5.89$, MSE = 0.51, p = 0.026, $\eta^2 = 0.25$) suggesting that both groups of participants felt a loss of control when returning to MSW2K. The statement being rated reflects a participant's general sense of control over the software and not simply their control of the menus and toolbars.

Summary: The analysis suggests that at the outset the feature-shy participants felt that they were less in control of the MSW2K software than did the feature-keen participants, however, the feature-shy participants experienced an increase in control with MSW Personal. The feature-keen participants did not experience a change in control when they switched to MSW Personal. Both groups of participants appear to have experienced a loss of control when they switched back to MSW2K after having used MSW Personal for 4 weeks.

Hypothesis Supported: Partially. Feature-shy participants felt a better sense of control with MSWord Personal, but this was not the case for the feature-keen participants.

H8 Learnability Hypothesis. In the Q1 vs. Q6 comparison the MSWord version had a borderline significant main effect on learnability ($F(1,18) = 4.13$, MSE = 0.61, p = 0.057, $\eta^2 = 0.19$) showing that both groups of participants' perceived ability to learn the available functions was greater with MSW Personal than with MSW2K (Figure 8). Personality type also had a borderline significant main effect on learnability ($F(1,18) = 4.07$, MSE = 0.60, p = 0.059, $\eta^2 = 0.18$) showing that, independent of software version, feature-keen participants felt better able to learn the functionality offered than did the feature-shy participants.

The Q6 vs. Q7 comparison showed that the software version had a borderline significant main effect ($F(1,18) = 3.08$, MSE = 0.20, p = 0.096, $\eta^2 = 0.15$) whereby participants' perceived ability to learn decreased when they returned to MSW2K.

Summary: The analysis suggests that the feature-keen participants generally find it easier to learn functions than do the feature-shy participants, and that overall it was easier to learn functions with MSW Personal than with MSW2K.

Hypothesis supported: Yes.

H9 Three-way Comparison Hypothesis. In the final debriefing interview, participants were asked if they could explain how what they called the "changing menus" worked (MSW2K's adaptive menus). Although all participants were aware of the short and long menus and could explain how to expand the menus, 7 of the 20 participants (35%) had to be informed that the short menus were in fact adapting to their personal usage. Given our sample, which included no novice users, this was particularly suprising. Participants were then asked to rank according to preference MSW Personal, MSW2K with adaptive menus, and MSW2K without adaptive menus (the standard "all-in-one" style interface).

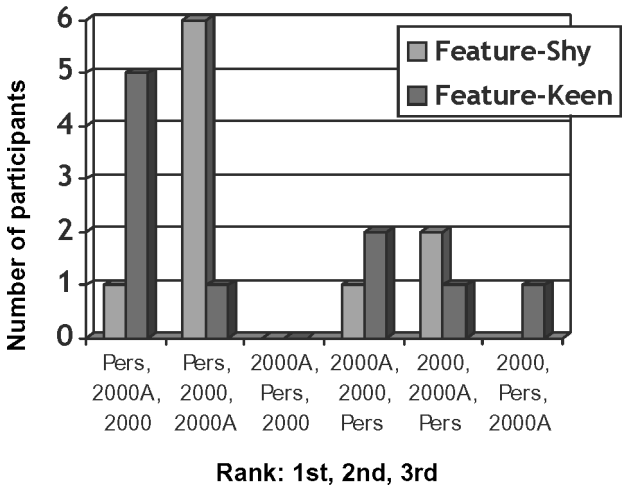

Fig. 9. Ranking three different interfaces for MSWord: Personal, 2000, and 2000 with adaptive menus (2000A) (N = 20).

Figure 9 shows the frequency of the three-way rankings. Of the six possible rankings, only five occurred. We analyzed the frequency with which each menu condition was ranked first, by calculating the Chi-square statistic to determine if actual frequencies were significantly different from the case in which all frequencies are equal. If we consider all 20 participants, there was a significant overall preference for MSW Personal (13 participants, 65%, $\chi^2(2, 20) = 9.10$, $p = 0.011$). We cannot apply the Chi-square statistic independently for the feature-keen and feature-shy because of our small sample sizes. Instead, we next describe the data for each group to indicate possible trends.

To make two-way comparisons between the interfaces for each of the personality types, we aggregated across the rankings. For example, by looking at the two leftmost ranking orders in the figure we see that 7 feature-shy participants preferred MSW Personal to the other two designs. From the remaining ranking orders we see that 3 feature-shy participants ranked the all-in-one design before the MSW Personal design. This shows that for the feature-shy there was preference for the MSW Personal to the all-in-one design: 7 participants to 3 participants. One can repeat the same steps to find that the feature-shy preferred the all-in-one to the adaptive design (8 to 2). However, the feature-keen did not prefer the all-in-one to both the adaptive and MSW Personal designs as expected. In fact, MSW Personal was preferred to adaptive (7 to 3) and preferred to the all-in-one (6 to 4) but the adaptive was preferred to the all-in-one (7 to 3).

Only 2 of the feature-shy ranked adaptive before all-in-one as compared to 7 of the feature-keen.

Summary: Although all participants were aware of the short and long menus in MSW2K, 35% had to be told that the contents of the short menus were adapting to their function usage. MSWord Personal was preferred by the majority of feature-shy and feature-keen participants, 65% of all

participants. Feature-shy's overall ranking was: adaptable, all-in-one, adaptive. Feature-keen's overall ranking was: adaptable, adaptive, all-in-one.

Hypothesis Supported: Partially. Feature-shy participants did rank adaptable, all-in-one, and then adaptive; but the feature-keen participants did not rank all-in-one before adaptive and adaptable.

Discussion and Additional Qualitative Feedback. Here we discuss our findings related to the comparison of the two interfaces. As before, we include participants' comments, both from the open-ended sections in questionnaires Q1 through Q8 and in the final debriefing interview, to provide more context for the quantitative results.

Adaptive Menus. The adaptive menus of MSW2K were liked by some and strongly disliked by many, but others had little opinion either way.

There were three participants who ranked the adaptive menus in Word 2000 first. Two had very positive comments when asked if they were aware of these menus and if they knew how they worked. For example:

Yes [I have noticed the "changing" menus], love that. It does it on my operating system as well... Yes [I know how they work], it seems that the functions that you use most often are the ones that show up. Or I don't know if they are the ones that I use most often or the ones that are used most often. I haven't figured that one out yet. .... Actually, I don't think it is the ones that I use most often. I think that it is a standard small set and then you click on the bottom and the whole set comes up. [Interview, Participant 22]

It seems like it just responds to whatever functions you use most recently. It gives you the most recent five or whatever. I like that kind of personalization because it is more dynamic and it just seems that I am always changing what I am doing from day to day. [Interview, Participant 10]

Interestingly, both participants were expert long-term users of MSWord and although they were aware of the adaptive menus, neither of them could fully explain how they worked. Participant 22 suspected that the menus adapted to her usage but then questioned whether this was right. Participant 10 knew that they were adapting but implied that there was a maximum number of items that could be shown when in fact if one had used all of the menu items recently, they would all appear in the short menu. This suggests that the user does not have to fully understand the conceptual model of an adaptive interface in order to be satisfied with the interface. If the adaptation "works well enough", then understanding the underlying mechanics is not important, at least for some users.

There were seven participants who had very negative experiences of the adapting menus. The first comment below refers to the ordering of items when the menu expands from the short version to the long one. If a desired item is not found in the short menu, then a user will likely have to rescan the full long menu because the newly-visible items are interspersed with those that have already been scanned. This usability problem might be fixed by highlighting the newly-visible items in such a way that they would be scanned first.

I don't know why [I dislike the adaptive menus], because I know that... well I have some idea why but... Well okay, first of all, part of the advantage of having these mega menus is that you can hunt through them and I realize that the stuff you use more tends to show up at the top. But when you click the open, the adaptive menus, the menu shows up in the way it is normally. So if there's—if you use two functions and they are right side by side and then there's actually a bunch of stuff in between, that will show up in between. And so I find it's like I have to start all over again looking through the menus for the functionality, which I find really annoying. And I don't know why. It's confusing, I just find it more confusing. I think that's ultimately it. And I don't think of myself as a naïve user and I don't know why it bothers me so much it just does. [Interview, Participant 3]

... [T]he adaptive menus are hell, I don't like them at all. So like that's a definite No—like that's almost a zero choice. I would never pick that, like I just hate it. [Interview, Participant 7]

I hate the menus where only your most recently used items show up first!!! [Q7, Participant 11]

Well the first thing is with the Word 2000—I really really really dislike the—I mentioned this with my questionnaire before—the frequency of use menu. I was often making mistakes and because they only give you, I don't know, a fixed number, maybe six menu items, I tend to use a lot of different functions regularly all of the time. So I was always, you know, using that little piece [down arrow icon], and I was always making a mistake going—where is it? Where is it? Where is it? It's gone! It fell off. I found that just... I still find that incredibly frustrating. So I would rather not do that and Word Personal didn't do that. So I much prefer it. [Interview, Participant 16]

Four additional participants felt negatively about the adaptive menus but not to the same extreme as the previous seven participants. For example:

I don't really feel one way or another about that. In fact I'd rather it didn't do that because sometimes I forget like I'm looking for something and I'm like—oh, I can't find it, where is it? And I can't find it because it's a hidden thingy. [Interview, Participant 6]

One participant did appreciate having only some options visible through the shortened menus but ultimately found that MSW Personal provided a better balance for him:

This feeling that you will forget that certain functions are there if you leave [the adaptive menus turned] on but also the menus are way too long if you leave everything on [i.e., adaptive menus turned off so that you have the full menu]. So it's a balance between the two. That's why the Personal gave me the balance I wanted. [Interview, Participant 17]

To summarize, 13 participants expressed opinions about the adaptive menus in MSW2K. For two participants, these menus worked very well. They were very strongly disliked by seven, and four were mildly negative. One possible explanation is that the adaptive model behind the menus provided a "better

fit" for the usage patterns of the two satisfied participants than for the other participants' usage patterns. Understanding the required "degree of fit" of an adaptive interface in order to achieve usability is an area of future investigation.

Individual Differences in Satisfaction and Control. Unlike the results from the first set of hypotheses, a number of differences between the feature-shy and the feature-keen participants are suggested in the self-reported measures from Q1 through Q7. Results for perceived control and satisfaction were dominated by interactions, whereby feature-shy participants experienced an increase in both satisfaction and control while using MSW Personal and the feature-keen did not experience any significant difference. One way this can be interpreted is that MSW Personal appears to have improved satisfaction and sense of control for the feature-shy without negatively affecting the feature-keen. Once they had used MSWord Personal for 4 weeks, the feature-shy were able to achieve a comparable level of satisfaction and perceived control to the feature-keen. This suggests that through the redesign of the user interface we can improve the experience of one group of users without negatively affecting the experience of another group.

Navigation

The comparison of the Q1 and Q6 data showed a strong effect of navigation, for both the feature-shy and the feature-keen. Some of our participants specifically noted the time savings when there were fewer options to navigate through in the menus and toolbars. Representative comments include:

I really like having only the tools I use very frequently on my interface if I so choose. It makes me more efficient as I don't have to look around for the function I need. [Q2, Participant 15]

While I use a standard set of features for most of my work, I am pleasantly surprised when I go to use a feature I haven't used for a while and find it's the only one in the menu. It makes my task faster and less frustrating. [Q5, Participant 12]

I'd be in the Microsoft Word interface and it'd be like—oh God just too many buttons. Like I don't think that Microsoft does a really good job of making their icon match what the button actually does. And I will sit there and I will have to hover over the button and wait for the explanation to come up. And it's like oh man, what a waste of time! So that's when I'd find myself getting like, okay, I don't need all this crap right now. It's too hard to find things on all the menubars and that's when I'd switch back to the personal. [Interview, Participant 11]

These comments suggest that the difference in navigation between MSWord Personal and MSW2K is not a subtle difference.

Learnability. When participants were asked to assess the learnability of the multiple-interfaces design and the adaptive design in questionnaires Q1 to Q7, the multiple-interfaces design had significantly higher ratings. In the debriefing interview, however, the all-in-one style interface was presented as an option alongside the other two interfaces.

Fifteen participants indicated that the all-in-one interface was best and the standard reason given was that seeing all of the menu items all of the time gives one a sense of what is available and thereby promotes learning the available functions. Some participants specifically mentioned learning through exploration. Two representative comments are:

2000 without the changing menus [is best] just because you can see all of your options so you know what all of the features are. [Interview, Participant 2]

I want to learn them all or nothing. . .In general I think that if they are there you are more likely to say—what is this?—and use it. So maybe a little bit. Because I have explored. The only way I've learned to use the program is by playing with it. So I saw the indexes and I went—how do you do that?—so now I know how to make an index. I guess if I never saw it, I'd probably never have played with it. [Interview, Participant 22]

Two participants indicated that having everything available in the Full Interface within MSW Personal supported learning equally as well as did the all-in-one interface. They did not need to be accessing the full menus all of the time:

I think if I can switch to the full interface like that, it's very convenient. So I think the learning ability [in MSW Personal] shouldn't be impacted. . . . Just one click. [Interview, Participant 5]

I want it all or I want mine [personal interface]. In the same way, I don't want the computer deciding what it's going to show me. I want to decide myself. If I don't know how to do something then I want to go and use the full interface more as like a reference or something and have it all kind of there. [Interview, Participant 11]

Two participants indicated that they learn through exploration but that they are not in exploratory mode all of the time. Having a Personal Interface forced them to take ownership of learning as they actively decided when to enter exploratory mode by switching to the Full Interface. For example:

I think the one with the adaptive menus doesn't support it [learning] at all because it just disappears on you and you don't even know that it is there. I would say that it is probably similar between the regular Word long menu and the Personal one. Because you still have to think that you need something different and find it. Often my strategy around that is that somebody says—Oh, try this—or—there must be a way to do this—and then go to help or whatever. [What about learning by remembering labels you have seen in the long menus?] *That's not been my experience myself. . .I must say that whether it's the Personal sort of thing or the long menus, for me at least it's—oh, I have to go exploring, I'm going to go look. Because with your sort of routine daily functions I am not using the menus, I'm not paying attention to the menus. So I'm not in explore mode. I'm not even in attentive mode, I wouldn't even notice if I've seen something related or not. So I wouldn't usually notice.* [Interview, Participant 16]

None of the participants thought that the adaptive menus best supported learning, which is a strong statement.

Understanding learnability is a rich area for further research. Our participants certainly did not perceive that minimalist interfaces provide scaffolding for learning; we saw no general perception of a Training Wheels effect [Carroll and Carrithers 1984]. The adaptive interface provides minimalist short menus, and they were ranked last for learnability by all of our participants. While five participants ranked the adaptable interface first, the majority thought that all-in-one best supported learning. One possible next step would be to evaluate the learnability afforded by these different interfaces with novice users, and to use objective measures of learning rather than self reports. We did not include any novice users in our study.

Cost/benefit Trade-off to Personalization. Personalization can be framed in terms of a cost/benefit trade-off.[9] The goal is that the cost of personalizing (time, attention away from primary task) will be outweighed by the benefits of personalizing. We have already mentioned some of the benefits, which include reduced navigation time. We note here that some of the participants were analogously very aware of the costs; for example, the cost to set up the Personal Interface:

The personal interface is an interesting concept but seems time consuming to completely set up. I am still adding features to it. [Q3, Participant 18]

Initial configuration was time-consuming but it is ok if it only must be done once. [Q8, Participant 16]

Another cost is the additional complexity added to the interface. While the goal of multiple-interfaces is to allow users to work predominantly in a simplified interface, there is additional functionality that needs to be included in the interface in order to make this possible. For some, that cost dominated:

The thing is for me was that I want my software to be pretty much invisible to me and the personal required more visibility than I would have liked it. [Interview, Participant 13]

For others, however, the benefit clearly dominated:

I think something like that [MSW Personal] should be made available. It's a nice thing—it's a nice interface. I mean, you know, I don't know how easy it would be to be available to many people. I guess that you would have to package it or whatever. But it was a nice addition. I actually enjoyed it. [Interview, Participant 15]

I think that Word XP[10]*needs a personal edition even more than 2000.* [Q8, Participant 17]

I would like to have Personal re-enabled on my machine! [Q8, Participant 16]

Thus, for some users, even if the cost to personalizing was relatively high, there was sufficient benefit derived from a Personal Interface to make it

[9]Mackay, used an economic analogy: when a user takes time to customize the user is trading off a short-term investment of time for a longer-term potential gain in productivity [1990, 1991].
[10]MSWord XP is the successor version to 2000.

worthwhile. For others, the cost outweighed the benefit. The difficulty for design is that the perceived cost and benefit are both dependent on individual users and difficult to determine a priori.

Overall Interface Preference. MSWord Personal was preferred by the majority of our participants, seven feature-shy and six feature-keen. Having such strong support by the feature-keen was unexpected. However, as noted above, the two groups of participants differed in their second ranking. Only two of the feature-shy ranked adaptive before all-in-one as compared to seven of the feature-keen. This can perhaps be explained in part by the fact that six of the seven participants who were unaware of the adapting short menus were feature-shy participants. This indicates that lack of knowledge that adaptation is taking place likely contributes to overall dissatisfaction with an adaptive interface.

Interestingly, of the 13 participants who expressed an opinion about the adaptive menus beyond a simple ranking, only two were positive. Yet three participants ranked adaptive first and 9 ranked it second, so adaptive menus did have supporters. The imbalance in the comments about these menus suggests that those who disliked the menus had a more extreme or passionate dislike as compared to those who liked the menus. The implication for user interface design could be that even if a design works sufficiently well for a large part of the user population, if that same design is perceived by others in the user population to work very poorly, the negative views will dominate.

Not surprisingly, the group of 13 participants who ranked Personal first is almost identical to the group of 13 who did not give up on their desired approach to personalization. (One participant who didn't give up did not rank MSW Personal first and one participant who did give up did rank it first.) Given that this group spanned the possible personalization strategies, it suggests that flexibility of personalization played a role in the interface ranking. Users have the ability to personalize when they want and what they want in MSWord Personal. There is no such flexibility in MSW2K, which implements a single personalization strategy.

Independence of Variables. One might argue that our dependent measures of satisfaction, navigation, control, and learning are at least somewhat related to our independent variable of personality type (feature-keen and feature-shy). For example, it may not be entirely surprising that the feature-shy were significantly less satisfied than the feature-keen with MSW2K at the time that Q1 was administered. After all, MSW2K is a feature-rich application. There was another independent variable, however, namely the two interface conditions. It is the impact of those conditions on the dependent measures that is most interesting in the findings we have reported in this section.

The results of this evaluation are promising, however, there were inherent limitations and constraints to the experiment design that may have affected the results. Four threats to validity in our experiment are:

(1) **Reactive effect:** Participants were fully aware of their participation in the experiment and some may have adjusted their interactions and responses.

(2) **Multiple-treatment interference:** Participants were exposed to two versions of MSWord and there were likely effects of having used one version that were not erasable when using the second.
(3) **Researcher interference:** A single researcher performed the role of the experimenter for this experiment. There may have been something specific to the particular researcher that systematically affected the results.
(4) **Selection bias:** Participants were a self-selected group because we did not have a sampling frame of all MSW2K users and therefore could not draw a simple random sample.

The best way to ensure that there wasn't anything incidental in our experiment execution that determined the results would be to replicate the experiment. Ideally we would want to conduct a longer field experiment with a different person acting in the researcher role. Counterbalancing the order in which software versions are used would be ideal. In addition, using a different application, whether it be another word processor or another general productivity application, would go a long way to ensuring the generalizability of the results to the class of word processing applications or general productivity applications as a whole.

8. CONCLUSIONS AND FUTURE WORK

We conclude with some final thoughts about MSWord Personal, adaptive versus adaptable designs, individual differences, user assisted personalization, and other scenarios of use for multiple interfaces designs.

MSWord Personal. The multiple-interfaces design of MSWord Personal performed very well in our field evaluation. Unlike previous work by Mackay [1991], which showed that users customize very little, the majority of our participants did personalize and they did so according to their function usage. The fact that MSW Personal offers a new style of interface, unfamiliar to all our participants, and requires effort to use, did not preclude the majority of participants (65%) ranking it first, preferring it to both the adaptive interface of MSW2K or an all-in-one style interface. We expect that had it been possible to add functions faster, even more participants would have ranked Personal first. That almost as many of the feature-keen (6 participants) as the feature-shy (7 participants) ranked Personal first is particularly encouraging.

Adaptive vs. Adaptable. Despite the breadth of research into adaptive user interfaces, there has been relatively little empirical comparison between adaptive and adaptable interfaces, and to date all investigations have been relatively short laboratory experiments (e.g., Debevc et al. [1996]). Our experiment allowed us to compare one instance of each of these design alternatives in the context of a commercial software application with users carrying out real tasks in their own environment. Results favored the adaptable design but the adaptive interface definitely had support.

The adaptable design implemented in MSWord Personal combines several design elements: two interfaces (one personal interface, one full interface), a

simple toggle to switch between the interfaces, an easily adaptable personal interface under full user control, and a small initial personal interface. The interface of MSW2K, by comparison, has a single interface, which is adapted solely by the system, with the exception that the user can easily open a short menu into a long menu. Our evaluation did not isolate the effects of the different interface design elements, although where possible we did get qualitative feedback on those elements. Our findings suggest that MSW Personal was preferred to MSW2K because user-controlled interface adaptation results in better navigation and learnability, and allows for the adoption of different personalization strategies, as compared to system-controlled interface adaptation, which implements a single strategy.

Because there were several differences between the two conditions compared, we do not assert that two interfaces are always better than one, nor that adaptable is always better than adaptive. A 2×2 experimental design, comparing one/two interfaces by adaptive/adaptable, would be required to tease this apart. We did not do this, for the reasons given earlier. Based on the qualitative feedback in our evaluation, however, we strongly believe that the two-interface aspect of MSW Personal was a key contributing factor to its success; it allowed users the flexibility to adopt different personalization strategies. We did in fact observe different personalization strategies.

We note that the effect of the dual interface, namely support to easily move back and forth between a personalized interface and an interface with default settings, could have been achieved through a single-button "factory reset" operation if the reset was undoable at any time by the user. We believe that conceptually this model would be more difficult for users, especially novices, to understand and trust. Informative next steps in the research include comparing only two-interface designs, one where the personal interface is under user control and the other where it is under system control, as well as comparing a two-interface design to a factory reset model that achieves the same outcome but with a different conceptual model for the user.

More recent laboratory research investigating adaptive designs shows conflicting evidence. Findlater and McGrenere [2004] found that adaptive split menus [Sears and Shneiderman 1994] were slower than static split menus, and slower than adaptable split menus in most circumstances. Subjects also preferred the adaptable menus to both the static and adaptive ones. Gajos et al. [2006] found an adaptive interface to be faster than a static one, but no adaptable alternative was evaluated. They also suggest some reasons for the conflicting evidence, for example, the frequency with which adaptations occur, but more work is needed to tease these issues apart. Alpert et al. [2003] investigated user attitudes regarding a user-adaptive e-commerce website and found that users were unenthusiastic toward system attempts to infer user needs and provide adapted content accordingly. A strong desire to have full and explicit control of the content and interaction was expressed. Jameson and Schwarzkopf [2002] compared adaptable and adaptive systems for adding items to a hotlist for a conference web site. While there was no performance difference, anecdotal evidence showed that some subjects strongly preferred the adaptive system, while others strongly preferred the adaptable.

Individual Differences. The existence of individual differences with respect to software features is an idea that has been proposed in the literature [Kaufman and Weed 1998; McGrenere and Moore 2000] but has undergone minimal evaluation. Based on our research it appears to have construct validity. One of the most interesting observations is that while there were no substantial differences between the feature-keen and the feature-shy participants in terms of how they used the two interfaces and how they approached the task of personalizing, there were a number of differences observed in terms of the self-reported measures. The feature-shy felt more satisfied and experienced a greater sense of control with MSW Personal than with MSW2K, whereas there were no differences detected for the feature-keen on these measures. Further work is required to validate the instrument used to assess the individual differences and to understand how this aspect of personality relates to other well-documented personality differences.

User-Assisted Personalization. In order to shift the cost benefit ratio of personalization, one needs to increase the benefit and decrease the cost. Benefit can be increased by ensuring that the personalized interface is always a "good fit" for the user, and that the cost is minimal. One way to achieve both of these is for the system to assist the user in personalizing: the system provides adaptation suggestions based on usage information and allows the user to accept or reject the suggestions. This moves the design in the direction of user-assisted personalization that relies on user-modeling technology (this was sometimes called *user-controlled self adaptation* in the early literature [Dieterich et al. 1993] and more recently *mixed-initiative* interaction [Horvitz 1999]). The advantage of this approach is that the user retains ultimate control and the system does the bookkeeping to manage the knowledge of function usage and changing patterns in usage, a task at which the system is particularly adept. This approach has been investigated by others, for example, in Flexcel [Krogsoeter et al. 1994] and the adaptive toolbar [Debevc et al. 1996], both with mixed user testing results. This research was conducted over 10 years ago but has not been commercialized. It has recently resurfaced in the research community [Lim et al. 2005; Miah et al. 1997], suggesting that further exploration of mixed-initiative interaction is underway. Key aspects of ongoing research will be to inform *how* and *when* to provide personalization suggestions. In terms of *when*, we know from the current study that there was both a strong initial trigger to add functions, and a need to amortize the cost of customizing an immediately-needed function by, at the same time, adding functions that are expected to be used in the future. These trigger points would be naturally occuring user behaviors upon which user-assisted personalization research could build.

Other Scenarios for Multiple Interfaces. We believe the concept of multiple interfaces has potential beyond the level-structured design seen today in some commercial applications. A possible scenario of use for multiple interfaces is to support users making the transition to a new version of an application; for example, MSW2K could include the MSW 97 interface. Often users delay upgrading their software because of the time required to learn a new version. By allowing users to continue to work in their old interface with single-button

access to a new interface, users would be able to transition at a self-directed pace. Multiple interfaces might also be used to mimic a competitor's interface in the hopes of attracting new customers; for example, MSWord could offer the full interface of a different word processor such as WordPerfect (or vice-versa) with a single button click, in order to support users making the transition to a new product.

In all three scenarios, help facilities could take advantage of the fact that both interfaces are accessible to show the user how functions in one interface map to functions in the second interface. This is a good example of how the adaptable nature of a multiple interface design leaves the user more in control of the interface. We believe this is especially important during critical transitions such as from novice to experienced user, from one version of a product to the next, and from one product to a competing or replacement product. There are of course other interface differences beyond menus and toolbars that need to be considered for the multiple-interface paradigm. This too is an area for future work.

ACKNOWLEDGMENTS

We are grateful to Mary Czerwinski for her assistance with both the design of the study and the statistical analysis. Microsoft Corporation provided the logging technology and expertise questionnaire. The conceptual design of MSWord Personal originated from joint research with Gale Moore within her Learning Complex Software Project. Funding was provided by IBM Canada through a graduate fellowship for Joanna McGrenere and by the Natural Science and Engineering Research Council of Canada. We are also grateful for all the helpful comments and suggestions from all the anonymous reviewers.

REFERENCES

ALPERT, S. R., KARAT, J., KARAT, C.-M., BRODIE, C., AND VERGO, J. 2003. User attitudes regarding a user-adaptive eCommerce website. *User Model. User-Adapt. Interact. 13*, 373–396.

BROWNE, D., TOTTERDELL, P., AND NORMAN, M. EDS. 1990. *Adaptive User Interfaces*. London: Academic Press Ltd., London, UK.

CAMPBELL, D. T. AND STANLEY, J. C. 1972. *Experimental and Quasi-Experimental Designs for Research*. Rand McNally & Company, Chicago, IL.

CARROLL, J. AND CARRITHERS, C. 1984. Blocking learner error states in a training-wheels system. *Human Fact. 26*, 4, 377–389.

COTE-MUNOZ, J. A. 1993. AIDA—An adaptive system for interactive drafting of CAD applications. In *Adaptive User Interfaces: Principles and Practice*. M. Schneider-Hufschmidt, T. Kuhne, and U. Malinowski, Eds., North-Holland: Elsevier Science Publishers B.V., Amsterdam, The Netherlands.

CSINGER, A., BOOTH, K. S., AND POOLE, D. L. 1994. AI meets authoring: User models for intelligent multimedia. *Artifi. Intel. Rev. J. 8*, 3, 447–468.

CYPHER, A. 1991. Eager: Programming repetitive tasks by example. In *Proceedings of ACM CHI'91*. ACM, New York, 33–39.

DAVIS, J., DYE, J., JOHNSON, N., AND BELL, S. 1999. *Microsoft Usability Report.*

DEBEVC, M., MEYER, B., DONLAGIC, D., AND SVECKO, R. 1996. Design and evaluation of an adaptive icon toolbar. *User Model. User-Adapt. Interact. 6*, 1, 1–21.

DIETERICH, H., MALINOWSKI, U., KÜHME, T., AND SCHNEIDER-HUFSCHMIDT, M. 1993. State of the art in adaptive user interfaces. In *Adaptive User Interfaces: Principles and Practice*, M. Schneider-Hufschmidt, T. Kuhme, and U. Malinowski, Eds., North Holland: Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 13–48.

FINLATER, L. AND MCGRENERE, J. 2004. A comparison of static, adaptive, and adaptable menus. In *Proceedings of ACM CHI 2004*. ACM, New York, 89–96.

FISCHER, G. 1993. Shared knowledge in cooperative problem-solving systems—integrating adaptive and adaptable components. In *Adaptive User Interfaces: Principles and Practice*, M. Schneider-Hufschmidt, T. Kuhme, and U. Malinowski, Eds., North Holland: Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 49–68.

GAJOS, K., CZERWINSKI, M., TAN, D., AND WELD, D. 2006. Exploring the desing space for adadptive graphical user interfaces. In *Proceedings of AVI '06*, 201–208.

GANTT, M. AND NARDI, B. 1992. Gardeners and gurus: Patterns of cooperation among CAD users. In *Proceedings of ACM CHI'92*. ACM, New York, 107–117.

GONG, G. AND SALVENDY, G. 1995. An approach to the design of a skill adaptive interface. *Int. J. Human-Comput. Interact. 7*, 4, 365–383.

GREENBERG, S. 1993. *The Computer User as Toolsmith: The Use, Reuse, and Organization of Computer-Based Tools*. Cambridge University Press, Cambridge, MA.

GREENBERG, S. AND WITTEN, I. 1985. Adaptive personalized interfaces—A question of viability. *Behav. Inf. Tech. 4*, 1, 31–45.

HORVITZ, E. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of ACM CHI'99*. ACM, New York, 159–166.

HSI, I. AND POTTS, C. 2000. Studying the evolution and enhancement of software features. In *Proceedings of International Conference on Software Maintenance*. 143–151.

JAMESON, A. AND SCHWARZKOPF, E. 2002. Pros and cons of controllability: An empirical study. In *Proceedings of Adaptive Hypermedia 2002*. 193–202.

KARAT, C.-M., BLOM, J., AND KARAT, J., EDS. 2004. *Designing Personalized User Experiences for eCommerce*. Kluwer Academic Publishers, Dordrecht, Netherlands.

KAUFMAN, L. AND WEED, B. 1998. Too much of a good thing? Identifying and resolving bloat in the user interface: A CHI 98 workshop. *SIGCHI Bulletin 30*, 4, 46–47.

KROGSOETER, M., OPPERMANN, R., AND THOMAS, C. 1994. A user interface integrating adaptability and adaptivity. In *Adaptive user support: ergonomic design of manually and automatically adaptable software*, R. Oppermann, Ed., Lawrence Erlbaum and Associates, Inc., Hillsdayle, NJ, pp. 97–124.

LANDAUER, T. 1997. Chapter 9: Behavioral research methods in human-computer interaction. In *Handbook of Human-Computer Interaction* (2nd ed.) M. G. Helander, T. K. Landauer, and P. V. Prabhu Eds., Elsevier Science B.V., Amsterdam, The Netherlands, pp. 203–227.

LIM, W. S., KIM, J. W., YOON, J. S., JANG, J. H., AND HAN, S. H. 2005. Usability of an adaptive toolbar in selecting functions. *J. Ergonom. Soc. Korea 24*, 4, 73–78.

LINTON, F., JOY, D., SCHAEFER, P., AND CHARRON, A. 2000. OWL: A recommender system for organization-wide learning. *Educat. Tech. Soc. 3*, 1.

MACKAY, W. E. 1990. Patterns of sharing customizable software. In *Proceedings of ACM CSCW'90*, ACM, New York, 209–221.

MACKAY, W. E. 1991. Triggers and barriers to customizing software. In *Proceedings of ACM CHI'91*, ACM New York, 153–160.

MACLEAN, A., CARTER, K., LOVSTRAND, L., AND MORAN, T. 1990. User-tailorable systems: Pressing the issues with buttons. In *Proceedings of ACM CHI'90*, ACM, New York, 175–182.

MALINOWSKI, U., KÜHME, T., DIETERICH, H., AND SCHNEIDER-HUFSCHMIDT, M. 1993. Computer-aided adaption of user interfaces with menus and dialog boxes. In *Proceedings of 5th Conference on Human-Computer Interaction*. 122–127.

MAYBURY, M. T. AND WAHLSTER, W. 1999. *Readings in Intelligent User Interfaces*. Morgan-Kaufmann Publishers, Inc., San Francisco, CA.

MCGRENERE, J. 2002. The Design and evaluation of multiple interfaces: A solution for complex software. Doctoral Dissertation, University of Toronto, Toronto, Canada.

MCGRENERE, J. AND MOORE, G. 2000. Are we all in the same "bloat"? In *Proceedings of Graphics Interface 2000*, 187–196.

MIAH, T., KARAGEORGOU, M., AND KNOTT, R. P. 1997. *Adaptive toolbars: An architectural overview*, from http://ui4all.ics.forth.gr/UI4ALL-97/miah.pdf

Microsoft Office 2000 Products Enhancements Guide. 2000. from http://www.microsoft.com/Office/evaluation/ofcpeg.htm

MILLER, J. R., SULLIVAN, J. W., AND TYLER, S. W. 1991. Introduction. In *Intelligent User Interfaces* J. S. Sullivan and S. W. Tyler Eds., ACM Press, New York, pp. 1–10.

NORMAN, D. 1998. *The Invisible Computer*. MIT Press, Cambridge, MA.

PAGE, S. R., JOHNSGARD, T. J., ALBERT, U., AND ALLEN, C. D. 1996. User customization of a word processor. In *Proceedings of ACM CHI 96*, ACM, New York, 340–346.

SCHNEIDER-HUFSCHMIDT, M., KUHME, T., AND MALINOWSKI, U. EDS. 1993. *Adaptive User Interfaces: Principles and Practice*. North-Holland: Elsevier Science Publishers B.V., Amsterdam, The Netherlands.

SEARS, A. AND SHNEIDERMAN, B. 1994. Split menus: Effectively using selection frequency to organize menus. *ACM Trans. Computer-Human Interact. 1*, 1, 27–51.

SHNEIDERMAN, B. 1997. *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (3rd ed.). Addison-Wesley Publishing, Reading, MA.

SHNEIDERMAN, B. AND MAES, P. 1997. Direct manipulation vs. interface agents: Excerpts from debates at IUI 97 and CHI 97. *Interactions 4*, 6, 42–61.

THOMAS, C. G. AND KROGSOETER, M. 1993. An adaptive environment for the user interface in Excel. In *Proceedings of ACM IUI '93*, 123–130.

VICENTE, K. J. AND TORENVLIET, G. L. 2000. The earth is spherical ($p < 0.05$): alternative methods of statistical inference. *Theoret. Issues Ergonom. Sci. 1*, 3, 248–271.

Received March 2004; revised August 2005 and July 2006; accepted July 2006 by Loren Terveen.