An Evaluation of a Multiple Interface Design Solution for Bloated Software

Joanna McGrenere[1,2], Ronald M. Baecker[1,2], and Kellogg S. Booth[3]

[1]University of Toronto
Department of Computer Science
10 King's College Road
Toronto ON, M5S 3G4
{joanna, rmb}@dgp.utoronto.ca

[2] University of Toronto
Knowledge Media Design Institute
10 King's College Road
Toronto ON, M5S 3G4

[3]University of British Columbia
Department of Computer Science
2366 Main Mall
Vancouver BC, V6T 1Z4
ksbooth@cs.ubc.ca

ABSTRACT

This study examines a novel interface design for heavily-featured productivity software. The design includes 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 the standard features. This design was prototyped as a front-end to a commercial word processor and evaluated in a comprehensive field study. The study 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 two-interface prototype with adaptable menus for the same word processor. Results showed that participants were better able to navigate through the menus and toolbars and were better able to learn with our prototype. There were also significant differences in satisfaction and control with our design.

Keywords

Featurism, bloat, personalization, adaptive/adaptable interfaces, individual differences, field study.

INTRODUCTION

Desktop applications such as the word processor, the spreadsheet, and the web browser have become woven into the daily lives of many people in the developed world. These applications have traditionally started "small" in terms of the functionality offered and have "grown" with every new release. This phenomenon, sometimes called creeping featurism [6,15] or bloatware [7], is pervasive: having a long feature list is now seen as essential for products to compete in the marketplace. Yet little attention has been paid to the impact of this functionality explosion on the user. Applications have also 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.

We describe an interface design that addresses the problem of featurism in productivity applications that are used by a diversity of users. The design has been instantiated as a front-end to the commercial word processor Microsoft Word 2000 (MSWord 2000) and evaluated in a field study that included 20 participants. Two main goals of the study were: (1) to assess the user's experience of the novel aspects of the interface design, and (2) to compare our design to a commercial interface design solution for heavily-featured software, namely the adaptive interface of MSWord 2000.

Design Solutions to Complex Software

The traditional "all-in-one" interface has menus and toolbars that are static and every user, regardless of tasks and experience, has the same interface. There are a number of alternative interface designs, in both the research literature and in commercial products, but there has been little or no evaluation of these designs, so their success is far from obvious.

Design solutions tend to fit into one of two categories: (1) ones that take a level-structured approach [16], and (2) ones that rely on some form of artificial intelligence. A level-structured design includes two or more interfaces, each containing a *predetermined* set of functions. The user has the option to select an interface, but not to select which functions appear in that interface. Preliminary research suggests, however, 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 [11]. 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 Light versions. Such product versioning, however, seems to be motivated more by business considerations than by an attempt to meet user needs.

The Training Wheels interface to an early word processor is a classic example of a level-structured approach that appears in the research literature. By blocking off all the functionality that was not needed for simple tasks, it was shown that novice users were able to accomplish tasks

CHI 2002, April 20-25, 2002, Minneapolis, Minnesota, USA.
Copyright 2002 ACM 1-58113-453-3/02/0004…$5.00.

significantly faster and with significantly fewer errors than novice users using the full version [2]. Despite the promise of this early work, the transition between the blocked and unblocked states was never investigated.

The broad goal of intelligent user interfaces is to assist the user by offloading some of the complexity [14]. *Adaptive* interfaces rely on intelligence to automatically adjust in a way that is expected to better suit the needs of each individual user. In practice this often results in the user perceiving a loss of control. *Adaptable* interfaces, in contrast, keep the user in control by providing the mechanisms for the user to personalize according to his/her needs [3]. There has been a debate in the UI community between those who promote intelligence in the interface and those who promote "comprehensible, predictable, and controllable interfaces that give users the sense of power, mastery, control and accomplishment" [17].

MSWord 2000 makes a significant departure in its user interface from MSWord 97 by offering menus that adapt to an individual user's usage [13]. When a menu is initially opened a "short" menu containing only a subset of the menu contents is displayed by default. To access the "long" menu one must hover in 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.

Two examples in the research literature that incorporate intelligence are Greenberg's work on Workbench that makes frequently-used commands easily accessible for reuse [4] and Linton, Joy, and Schaefer's *recommender* system that alerts users to functionality currently being used by co-workers doing similar tasks [9].

No user testing has been reported in the literature for any of the interfaces given above except for Training Wheels.

DESIGN OVERVIEW

Conceptual design

What makes our design unique is the *combination* of three design elements, rather than any single design element:

1) Two interfaces, one that is personalized (the Personal Interface), one that is the full set of functions (the Full Interface), and 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, therefore, unless the user adds many functions, it will remain a minimal interface relative to the Full Interface.

Note that the only difference between the two interfaces is the functions that are displayed visually in the menus and toolbars. The functions in the Personal Interface are effectively a subset of those in the Full Interface where the ordering of elements is preserved. Thus if the Full Interface contains a menu M with five items then the Personal Interface, if it contains M, will have at least one of the menu items and, if more than two menu items are included, the original ordering of items will be preserved. Thus, the only choice users make with respect to their Personal Interfaces is whether or not to include particular functions.

The idea of having more than one interface, with one that is personalized easily by the user, and putting the user in control of switching between interfaces was proposed by McGrenere and Moore [12] based on a study of 53 members of the general population who used MSWord 97. Their results showed that although a significant percentage of users are having a negative experience with the vast number of features in the interface, the majority of users would not choose a word processor that gave them only the functions that they are currently using. One reason for this is that users want the ability to discover new functions. This 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 upon application launch.

There are several reasons for having a personalizable interface (rather than a predetermined static small interface). Not only do users typically use very few features [9, 12], 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 [4]. Users also generally customize very little, likely because customization facilities are often powerful and complex in their own right and therefore require time both for learning and for doing the customization. (Mackay found the latter to be true in the case of UNIX customization [10].) This is a primary argument for an *adaptive* interface. Our goal has been to make an easy-to-understand *adaptable* interface instead.

Implementation

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

In order to evaluate this design in a field study with participants who were already users of MSWord 2000, our prototype was implemented so that it did not interfere with any customization that participants may have already made to their MSWord interface. It was also designed to be easily installed on top of an existing installation of

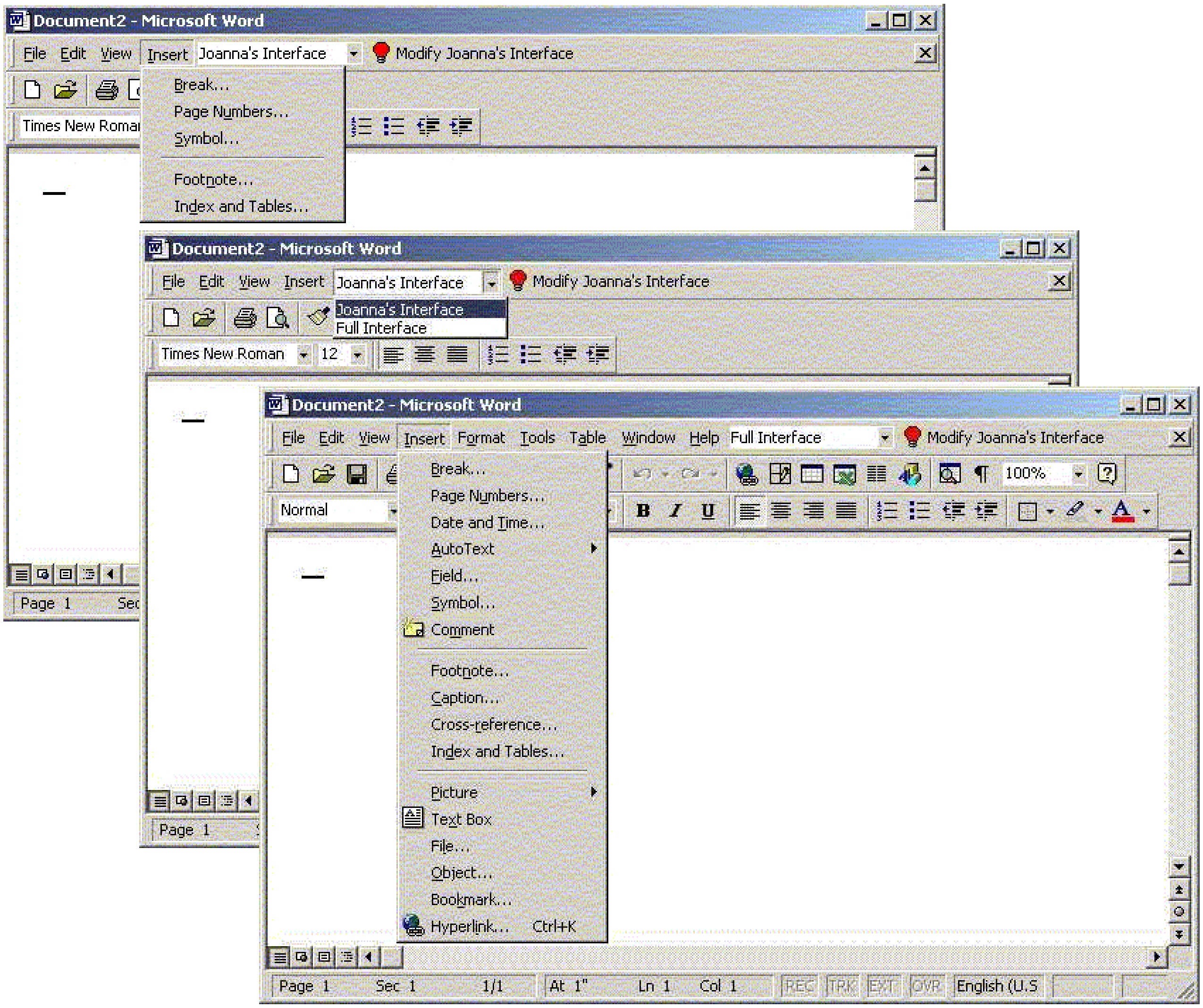


Figure 1: User opens the Insert menu in the Personal Interface, toggles to the Full Interface, and re-opens Insert Menu.

MSWord. This was accomplished by placing the required VBA code in a specialized document template that was loaded into MSWord on startup. If necessary, a user could have removed 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 with the software have occurred. Figures 1 and 2 show screen captures of the two interfaces as well as the personalizing mechanism.

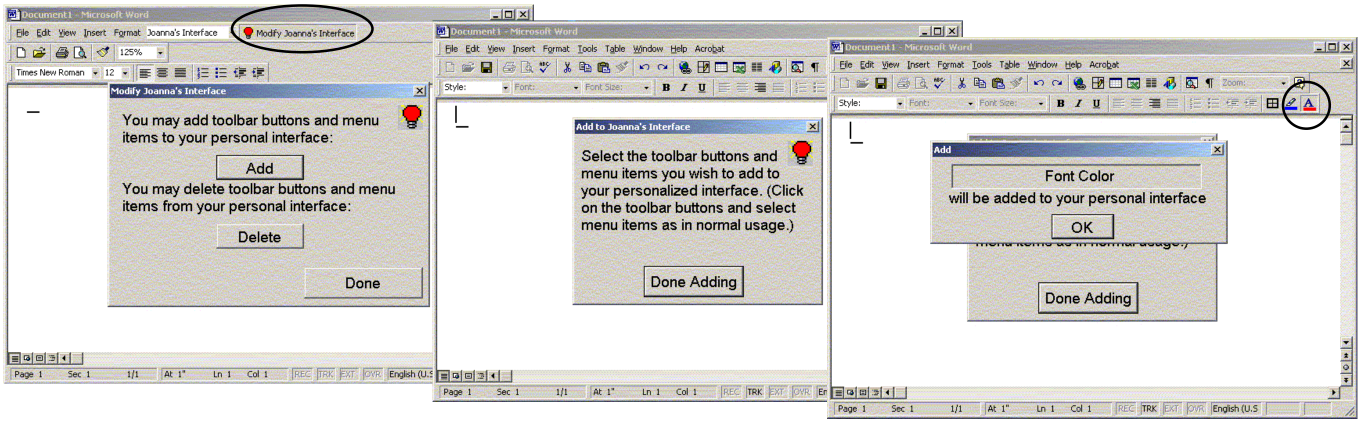
EVALUATION

Participants

Twenty intermediate and experienced MSWord users participated in this study. Participation was solicited electronically through newsgroups and listservs that broadcast across campus and the surrounding city. In order to participate users had to meet the following base criteria: they had to have been using MSWord 2000 for at least one month, had to do their word processing on a single computer, had to spend a minimum average of 3 hours on MSWord per week, had to have MSOffice expertise above the novice level, and had to be at least 20 years of age.

Personality/individual differences with respect to heavily-featured software were considered. We included 10 feature-keen participants and 10 feature-shy as assessed by an instrument developed by McGrenere and Moore [12]. 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 light version), and (3) the desire to have an up-to-date version of MSWord.

Prospective participants completed an online screening questionnaire and were considered in the order in which they applied. Table 1 provides a high level description of the participants. Education and expertise were rated on 7-point and 5-point Likert scales respectively.


Document1 - Microsoft Word
File Edit View Insert Format Joanna's Interface
Modify Joanna's Interface
Add
Delete
Done
File Edit View Insert Format Tools Table Window Help Acrobat
Add to Joanna's Interface
Done Adding
Add
OK

You may add toolbar buttons and menu items to your personal interface:

You may delete toolbar buttons and menu items from your personal interface:

Select the toolbar buttons and menu items you wish to add to your personalized interface. (Click on the toolbar buttons and select menu items as in normal usage.)

Font Color will be added to your personal interface

Figure 2: Process for adding a function to the Personal Interface – in this example the Font Colour function is added.

Table 1: Aggregate description of 20 participants.

	Gender		Mean age	Mean education (1 to 7)	Mean MSOffice expertise (1 to 5)	Mean # years using MSWord
	M	F				
Feature shy	3	7	late 20s	6.1	3.8	7.1
Feature keen	5	5	late 20s	5.6	4.5	7.0

Procedure

A field study was conducted instead of a laboratory study as it was expected that true personalizing behaviour would be significantly more likely to occur when users do their own tasks in their normal work context rather than in a lab setting with prescribed tasks.

Each participant was involved for approximately 6 weeks and met with the experimenter on 3 occasions as well as completing a series of short on-line questionnaires to assess experience with the software (Q1 – Q7). Refer to Figure 3 for an overview of the study protocol.

First Meeting: participant completed Q1 which assessed experience with MSWord 2000. Three things were then installed on the participant's machine (MSWord 2000 was already installed) – the prototype software which we called MSWord Personal, a software logger for capturing usage, and a small script to transfer the log files to a backup server on the Internet. Adaptive menus were turned off. Each participant's Personal Interface contained only 6 functions initially: 2 in the File menu and 2 on each of the Standard and Formatting toolbars.

Q2 through Q6: assessed MSWord Personal. Q2 was completed within two days of the First Meeting and was intended to capture the participant's first impression of MSWord Personal. Q3, Q4, Q5 and Q6 were completed one, two, three, and four weeks respectively from the First Meeting and were intended to capture the participant's experience of MSWord Personal over time.

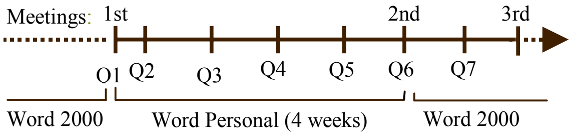


Figure 3: Schematic overview of the study protocol.

Second Meeting: held within one day of Q6 being completed. MSWord Personal was uninstalled leaving the participant with MSWord 2000 with adaptive menus on.

Q7: assessed participant's experience of MSWord 2000 one week following the Second Meeting.

Third Meeting: held two weeks following the Second Meeting. Participant's machine was completely restored to the state it was in prior to the study. A final semi-structured debriefing interview was conducted with each participant.

Instructions given to the participants: In advance of the study 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 – MSWord 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 it could be modified with the Modify button. The process of personalizing was not demonstrated. Participants were told that there was no right or wrong way to use the interfaces. They were specifically told 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 MSWord Personal would be uninstalled at the Second Meeting.

Schedule: 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 on the due date of each questionnaire with the participant's web page URL directly embedded in the email, facilitating 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.

Formal Design: the logistical constraints in conducting a field study precluded the counterbalancing of word processor conditions. The design is a 2 (personality types, between subjects) X 3 (levels, levels 1,3 = MSWord 2000, level 2 = MSWord Personal, within subjects) design where level 2 is nested with 5 repetitions. This design is best described as a quasi-experimental design [1].

Measures

The dependent measures were based on logging data and data collected from the 7 questionnaires. From the logged data we extracted the total time spent word processing, the time spent in each interface, the number of toggles between interfaces, and a trace of the modifications made to the Personal Interface. Each of the on-line questionnaires presented the same series of statements which were rated on a five-point Likert scale. The statements reflected overall satisfaction, ease of navigating through the menus and toolbars, control over MSWord, and ability to learn all the available features. The questionnaires also encouraged and provided space for open-ended comments.

Hypotheses

The hypotheses below are related to two of the main goals of the study: (1) to evaluate the user's experience of the MSWord Personal design, and (2) to compare our design to the adaptive interface of MSWord 2000.

Usage Hypothesis: The majority of the participants will choose to use their Personal Interface – they will use the personalizing mechanism to include all frequently-used functions and will spend the majority of their time in their Personal Interface.

Good Idea Hypothesis: The concept of having two interfaces will be easily understood and will be considered a good idea. The toggling mechanism will be considered an easy way to switch between the interfaces.

Satisfaction Hypothesis: Feature-shy participants will be more satisfied with MSWord Personal than with MSWord 2000.

Navigation Hypothesis: Both feature-shy and feature-keen participants will feel that they are better able to navigate the menus and toolbars with MSWord Personal than with MSWord 2000.

Control Hypothesis: Both feature-shy and feature-keen participants will feel a better sense of control with MSWord Personal than MSWord 2000.

Learnability Hypothesis: Feature-shy participants will feel that they are better able to learn the available features with MSWord Personal than with MSWord 2000.

Quantitative Results

In general, all six hypotheses were either fully or partially satisfied. For technical reasons we are missing some of the logging data for one of the participants – where this is relevant we note N=19, otherwise one can assume N=20.

Usage Hypothesis: The majority of participants did make use of the Personal Interface. During the four weeks that MSWord Personal was used, 14 out of 19 participants spent 50% or more of their word processing time in their Personal Interface (Table 2) and these same participants added all frequently-used functions (those menu and toolbar items used at least half of the days that the word processor was used).

Table 2: Number of participants by amount of time in their Personal Interface (N=19).

	% of time in Personal Interface				Total
	≥ 0%	≥ 25%	≥ 50%	≥ 75%	
Feature shy	2	2	2	4	10
Feature keen	1		3	5	9
Total	3	2	5	9	19

If we consider all participants, not only those who spent the majority of time in their Personal Interface, we can see that the great majority of functions that were used even 25% of the time or more were added to the Personal Interface (Table 3). For example, participants had on average 7.2 functions that were used between 25% and 50% of the days that word processing occurred and on average participants added 90% of these functions.

Table 3: Mean number of functions used with the given regularity. The number in brackets gives the mean percentage of these functions that were added to the Personal Interface (N=19).

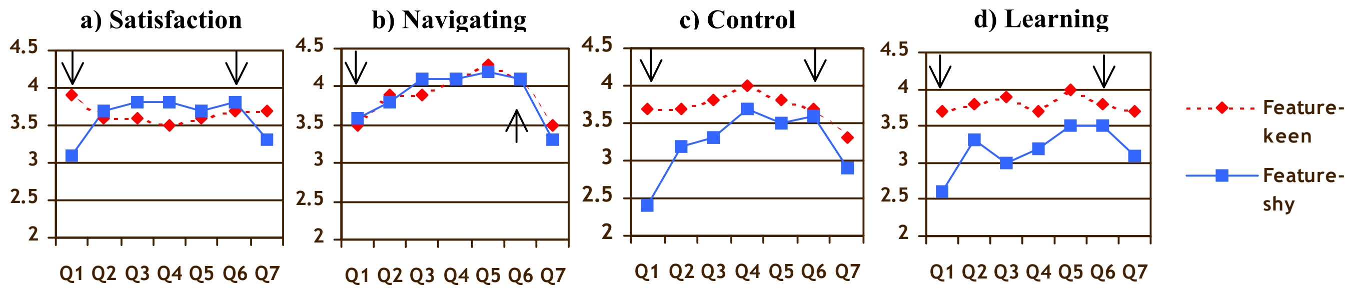
	Frequency of Function Use			
	> 0%	≥ 25%	≥ 50%	≥ 75%
Feature shy	31.2 (59%)	6.8 (88%)	3.8 (94%)	1.1 (100%)
Feature keen	28.2 (73%)	7.7 (92%)	2.4 (100%)	0.3 (100%)
Total	29.8 (66%)	7.2 (90%)	3.2 (96%)	0.7 (100%)

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

Together this data indicates that participants were capable of personalizing according to their individual function usage, it was easy to do, and the Personal Interface was actively used.

Good Idea Hypothesis: The concept of having two interfaces was easily understood (mean rating 4.4 out of 5) but having two interfaces being a good idea only received a mean rating of 3.9. We discovered through the debriefing interview with participants that the desirability of having two interfaces had been interpreted in two ways, namely that having both a Personal Interface and the Full Interface is better than (1) having just one interface with everything (i.e, a full interface), or (2) having just a Personal Interface. We had intended the first meaning.

The toggling mechanism had a mean rating of 4.5 out of 5 for its ease of use, indicating that a toggle is an effective way to switch between more than one interface.



"This software is satisfying to use."

"Navigating through the menus and toolbars is easy to do."

"It's easy to make the software do exactly what I want."

"I will be able to learn how to use all that is offered in this software."

Figure 4: Satisfaction, navigating, control, and learning. Graphs and original statements are given.

In order to assess the remaining hypotheses a series of factorial ANOVAs was run. We evaluated each of the four dependent measures (satisfaction, navigation, control, and learning) in three different analyses and the results are shown in Table 4.

1) Q1 vs. Q6: compares measures after extended time in each condition. Q1 responses reflect usage of one month or more with MSWord 2000. Q6 reflects one month's use of MSWord Personal.
2) Q6 vs. Q7: compares measures as an initial reaction of returning to MSWord 2000 after one month's use of MSWord Personal.
3) Q2, Q3, Q4, Q5, Q6: compares measures at regular intervals during four-week usage of MSWord Personal.

Table 4: All F values have degrees of freedom F(1,18) except for those in the shaded region which are F(4,72).

		Independent Variables	
Q1 vs. Q6	**Version (V)**	**Personality (P)**	**V X P**
Satisfy	1.27	1.12	4.12 **
Navigate	5.76 ***	.03	.05
Control	4.38 **	6.21 ***	4.38 **
Learn	4.13 **	4.07 **	2.64
Q6 vs. Q7			
Satisfy	.85	.18	.85
Navigate	8.02 ***	.07	.16
Control	5.89 ***	.70	.44
Learn	3.08 *	1.33	1.11
Q2 – Q6			
Satisfy	.27	.28	.27
Navigate	2.38 *	.00	.41
Control	2.02	2.32	.64
Learn	1.56	1.90	1.10

*p<.10 **p<.06 ***p<.05

In addition to reporting statistical significance we report 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. Landauer notes that effect size is often more appropriate than statistical significance in applied research in Human-Computer Interaction [8]. The metric for interpreting eta-squared is: .01 is a small effect, .06 is medium, and .14 is large.

Satisfaction Hypothesis: The MSWord versions had a different impact on the satisfaction of the two groups of participants (Figure 4a). There was a significant cross-over interaction for Q1 vs. Q6 ($F(1,18) = 4.12$, $p<.06$, $\eta^2 = .19$) prompting us to test the simple effects for each group of participants independently. The comparison was not significant for the feature-keen participants, however, the increase in satisfaction was borderline significant for the feature-shy ($F(1,9) = 3.645$, $p<.10$, $\eta^2 = .29$). Interestingly, there was no significant change in satisfaction when participants returned to MSWord 2000.

Navigation Hypothesis: The version of MSWord had a significant main effect on users' perceived ability to navigate in both the Q1 vs. Q6 comparison ($F(1,18) = 5.76$, $p<.05$, $\eta^2 = .24$) and the Q6 vs. Q7 comparison ($F(1,18) = 8.02$, $p<.05$, $\eta^2 = .31$) (Figure 4b). Both comparisons favoured MSWord Personal. There was also a borderline significant learning effect in Q2 through Q6 ($F(4,72) = 2.38$, $p<.10$, $\eta^2 = .12$) indicating that navigation became easier over time, however, none of the posthoc pairwise comparisons with the Bonferonni error correction were significant.

Control Hypothesis: The results of the Q1 vs. Q6 comparison of control are dominated by a significant interaction ($F(1,18) = 4.38$, $p<.06$, $\eta^2 = .20$) (Figure 4c). Testing the simple effects found the comparison to be non-significant for the feature-keen participants, however, the feature-shy perceived a significant increase in control ($F(1,9) = 11.17$, $p<.01$, $\eta^2 = .55$). There was a main effect for control from Q6 to Q7 ($F(1,18) = 5.89$, $p<.05$, $\eta^2 = .25$) indicating that both groups of participants felt a loss of control when returning to MSWord 2000. Note that the statement being rated reflects a user's general sense of control over the software and not simply their control of the menus and toolbars.

Learnability Hypothesis: In the Q1 vs. Q6 comparison the MSWord version had a significant main effect on learnability ($F(1,18) = 4.13$, $p<.06$, $\eta^2 = .19$) showing that users' perceived ability to learn the available functions was greater with MSWord Personal than with MSWord 2000 (Figure 4d). Personality type also had a significant main effect on learnability ($F(1,18) = 4.07$, $p<.06$, $\eta^2 = .18$) 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 ($F(1,18) = 3.08$, $p<.10$, $\eta^2 = .15$) whereby participants' perceived ability to learn decreased when they returned to MSWord 2000.

To summarize, both groups of participants favoured MSWord Personal in terms of their ability to navigate the menus and toolbars and to learn the features. In general, feature-keen participants felt they were better able to learn as compared to the feature-shy. Results for control and satisfaction were dominated by interactions – feature-shy participants experienced an increase in both satisfaction and control while using MSWord Personal and the feature-keen did not experience any significant difference. One way that this can be interpreted is that MSWord Personal improved satisfaction and sense of control for the feature-shy without negatively affecting the feature-keen.

Qualitative Results

Ranking Three Different Interfaces

In the final debriefing interview participants were asked if they could explain how the "expandable" (adaptive) menus worked. Seven of the 20 participants had to be informed that the short menus were in fact adapting to their personal usage. Participants were then asked to rank according to preference MSWord Personal, MSWord 2000 with adaptive menus, and MSWord 2000 without adaptive menus (standard "all-in-one" style interface). Figure 5 shows that 13 participants preferred Personal to either form of 2000. Aggregating across all of the feature-shy and feature-keen participants reveals an interesting difference: only 2 of the feature-shy ranked adaptive before all-in-one as compared to 7 of the feature-keen. This can perhaps be explained by the fact that 6 of the 7 participants who were unaware of the adapting short menus were feature-shy participants. This is an indicator that lack of knowledge that adaptation is taking place contributes to overall dissatisfaction with an adaptive application.

Design Suggestions

Although the personalizing mechanism in MSWord Personal (shown in Figure 2) was reported on the questionnaires to be easy to use, intuitive, and flexible, 3 participants commented about it being somewhat cumbersome. For example, the confirming dialog box that appears after the selection of each function was seen by some to be unnecessary. To counter this, none of the remaining 17 participants said that it was difficult to personalize. Our design goal for this mechanism was to make it easy to understand so we opted for a design that offered only basic functionality (adding and deleting functions) and that could be learned quickly through trial and error. One participant pinpointed our tradeoff: "The Add/Delete procedure seems slow and redundant for some reason, but is rather idiot-proof." This could be fixed by removing the confirmation dialog and designing a new form of menu that stays open and has check boxes adjacent to each item indicating its availability in the Personal Interface. This would eliminate the need to reopen the menu for each item that is added.

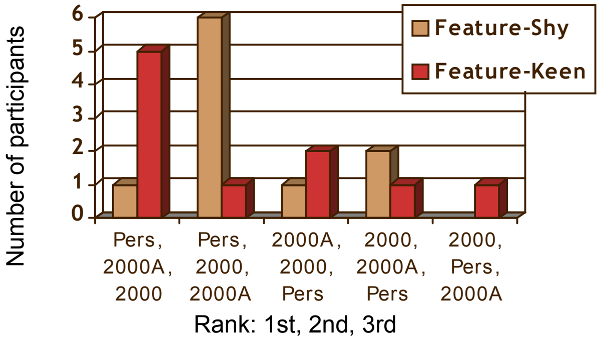


Figure 5: Ranking 3 different interfaces for MSWord.

Two participants felt that MSWord 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." Both of these participants were surprised when they were informed at the end of the study that this restructuring functionality is available through MSWord's native customize facility. Relative to our personalizing mechanism, this facility is very sophisticated and requires substantially more skill to use and to discover. Introducing this form of customization into the personalizing mechanism would likely make it inaccessible to non-advanced users.

Approach to Using Two Interfaces

Participants were not told how they should use the two interfaces in MSWord Personal. Of the 13 participants who ranked Personal first, 6 took the approach of adding all functions that they would ever use to their Personal Interface and then using it exclusively. The other 7 consciously chose to only keep the most regularly-used functions in their Personal Interface and preferred to switch to the Full Interface for additional functions. Of the 7 participants who did not favour Personal, 1 used the Full Interface almost 100% of the time and the other 6 would start in the Personal Interface and at some point would switch to the full one. These users expressed frustration with the time required to personalize and so they gave up on the personalizing process during the study.

LIMITATIONS AND CONSTRAINTS

The results of this study are promising and encourage further investigation in the design space of multiple interfaces. In parallel it would be prudent to replicate our current results in the context of a longer field study and with a different product to ensure that there wasn't anything incidental in our study that determined the results. Counterbalancing the order in which software versions are used would also be desirable.

CONCLUSIONS AND FUTURE WORK

The comparison between adaptive and adaptable interfaces has been mostly theoretical to date. This study has allowed us to compare one instance of each of these design alternatives in the context of a real software application with users carrying out real tasks in their own environment. Results favour the adaptable design but the adaptive interface definitely had support. Users were capable of personalizing according to their function usage and those who favoured a simplified interface were willing to take the time to personalize. One potential way of streamlining personalization would be to add a mechanism that provides usage information and allows the user to directly add features that have been used frequently or recently. This would move the design in the direction of user-assisted personalization (a mixed-initiative interface [5]), where the user has ultimate control but also benefits from user-modeling technology.

The existence of individual differences with respect to features is an idea that has been proposed [7,12] but has undergone minimal evaluation. Based on this research it appears to have construct validity. Further work is required to validate the instrument used to assess these differences and to understand how this aspect of personality relates to other well-documented personality differences.

The concept of multiple interfaces has potential beyond the level-structured design seen in some commercial applications today. One can imagine having multiple interfaces for a new version of an application; for example, MSWord 2000 could include the MSWord 97 interface. Some users delay upgrading their software because of the time required to learn a new version. By allowing users to continue to work in the old interface while also accessing the new interface, they would be able to transition at a self-directed pace. Similarly, multiple interfaces might be used to provide a competitor's interface in the hopes of attracting new customers; for example, MSWord could offer the full interface of a word processor such as Word Perfect (with single button access to switch between the two), in order to support users gradually transitioning to the Microsoft product. Clearly there are interface differences beyond menus and toolbars, so this would need further thought.

ACKNOWLEDGMENTS

We are grateful to Mary Czerwinski for her assistance with both the design of this 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 Sciences and Engineering Research Council of Canada.

REFERENCES

[1] Campbell, D.T., and Stanley, J.C. (1972). *Experimental and quasi-experimental designs for research*. Chicago, IL: Rand McNally & Company.

[2] Carroll, J., and Carrithers, C. (1984). Blocking learner error states in a training-wheels system. *Human Factors, 26(4),* 377-389.

[3] Fischer, G. (1993). Shared knowledge in cooperative problem-solving systems - integrating adaptive and adaptable components. In M. Schneider-Hufschmidt, T. Kuhme and U. Malinowski (Eds.), *Adaptive user interfaces: Principles and practice* (pp. 49-68). North Holland: Elsevier Science Publishers B.V.

[4] Greenberg, S. (1993). *The computer user as toolsmith: The use, reuse, and organization of computer-based tools*. Cambridge: Cambridge University Press.

[5] Horvitz, E. (1999). Principles of mixed-initiative user interfaces. *CHI 99*, 159-166.

[6] Hsi, I. and Potts, C. (2000). Studying the evolution and enhancement of software features. *International Conference on Software Maintenance*, 143-151.

[7] 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.

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

[9] Linton, F., Joy, D. and Schaefer, P., Charron, A. (2000). OWL: A recommender system for organization-wide learning. *Educational Technology & Society, 3(1).*

[10] Mackay, W. E. (1991). Triggers and barriers to customizing software. *CHI'91,* 153 – 160.

[11] McGrenere (2002). The design and evaluation of multiple interfaces: A solution for complex software. Doctoral dissertation, University of Toronto, Canada.

[12] McGrenere, J. and Moore, G. (2000). Are we all in the same "bloat"? *Graphics Interface 2000*, 187-196.

[13] Microsoft Office 2000 (2000). Products Enhancements Guide. http://www.microsoft.com/Office/evaluation/ofcpeg.htm

[14] Miller, J.R., Sullivan, J.W., and Tyler, S.W. (1991). Introduction. In J.S. Sullivan and S.W. Tyler (Eds.) *Intelligent User Interfaces* (pp. 1-10). NY: ACM Press.

[15] Norman, Don. (1998). *The Invisible Computer.* Cambridge, MA: MIT Press, 80.

[16] Shneiderman, B. (1997). *Designing the user interface: Strategies for effective human-computer interaction* (3rd ed.). Reading, MA: Addison-Wesley Publishing.

[17] 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.