Supporting Interface Customization using a Mixed-Initiative Approach

Andrea Bunt[1], *Cristina Conati*[1,2], *Joanna McGrenere*[1]

[1]Dept. of Computer Science,
University of British Columbia
Vancouver, BC, Canada
{bunt, conati, joanna}@cs.ubc.ca

[2] Dept. of Information and Communication Technology,
University of Trento
Povo, Trento, Italy

ABSTRACT

We describe a mixed-initiative framework designed to support the customization of complex graphical user interfaces. The framework uses an innovative form of online GOMS analysis to provide the user with tailored customization suggestions aimed at maximizing the user's performance with the interface. The suggestions are presented non-intrusively, minimizing disruption and allowing the user to maintain full control. The framework has been applied to a general user-productivity application. A formal user evaluation of the system provides encouraging evidence that this mixed-initiative approach is preferred to a purely adaptable alternative and that the system's suggestions help improve task performance.

ACM Classification: H.5.2 User Interfaces, *Graphical user interfaces*, *Evaluation/methodology*

General terms: Design, Human Factors, Experimentation

Keywords: mixed-initiative, adaptive, adaptable, GOMS analysis

INTRODUCTION

With every new release, software applications are packed with an increasing number of features. Feature-rich software applications have the potential to suit a wider range of individuals and thus are attractive from a marketing standpoint. This increase in functionality, however, has also been accompanied by an increase in the size and complexity of the graphical user interfaces (GUIs).

A wide variety of applications suffer from interface complexity, ranging from spreadsheet packages, to statistical analysis software, to image-editing software, all of which can have hundreds of features distributed throughout their menus and toolbars. These applications are often used by a diverse set of individuals, who differ not only in the types of tasks they wish to perform, but also in their application-specific knowledge and their general computer expertise. For the average user, this high degree of interface complexity translates into a visually cluttered interface, which can lead to both frustration [22] and decreased performance [1]. Even seemingly straightforward productivity software, such as word processors, suffer from these problems, as is evidenced by the number of researchers working on ways to assist users with this class of application (e.g., [7,9,16,19,21,24]).

Providing the user with a customized interface could mitigate the problem of interface complexity; however, how to best achieve an appropriately customized interface is a contentious issue (e.g., [25]). Two opposing approaches are *adaptable* and *adaptive*, which differ in terms of who is responsible for performing the customization (the system or the user) and consequently in the amount of control provided to the user. Adaptable interfaces permit full user control by providing users with interface mechanisms that allow them to customize their own interfaces. System-controlled adaptive interfaces, on the other hand, perform the customization automatically based on user-specific information, such as the user's work patterns and preferences. Since both approaches have their own benefits and drawbacks, the optimal solution likely lies somewhere in the middle. With adaptable interfaces, users are in full control, but not all users are willing to invest the effort necessary to customize [20] and some users may have limited ability to customize effectively [1]. Adaptive interfaces, on the other hand, do not require any extra effort from the user, but can suffer from a lack of user control, transparency and predictability [14].

In this paper we present the MICA (**M**ixed-**I**nitiative **C**ustomization **A**ssistance) system, which employs an innovative *mixed-initiative* approach, where the system and the user cooperate to produce a customized interface. Specifically, users are provided with an interface mechanism that gives them full control over the customization process, as well as adaptive support to help them customize their interfaces effectively. This approach is novel in that it is based on a form of cognitive modelling known as GOMS analysis [3], which provides MICA with two unique functionalities. First, MICA makes customization suggestions based on a principled and comprehensive quantitative assessment of how these suggestions are expected to impact the user's performance. To the best of our knowledge, no other work on interface customiza-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
IUI'07, January 28–31, 2007, Honolulu, Hawaii, USA.
Copyright 2007 ACM 1-59593-481-2/07/0001...$5.00.

Figure 1: The two-interface model: a Personal Interface (PI) is on the left, the Full Interface (FI) is on the right.

tion uses a formal quantitative assessment of performance to make informed decisions on how to customize. Second, MICA communicates these underlying expected performance savings to the users by providing access to its rationale, potentially improving the lack of transparency and predictability often present in adaptive interfaces. This also distinguishes MICA from other related systems.

In addition to MICA's framework, a second contribution of our work is a formal user study comparing the mixed-initiative approach to a purely adaptable alternative. The results of the study show that given an accurate user model, users prefer the support offered by the mixed-initiative system and the system's recommendations have a positive impact on user performance. The evaluation is a first step in validating the overall approach and is also one of the few evaluations to directly compare mixed-initiative and adaptable approaches to GUI customization.

RELATED WORK

While there are many examples of mixed-initiative systems (e.g., [15,26]), there has been little work on applying mixed-initiative approaches to the problem of GUI customization. The first exception involves systems that focus on a different form of customization: adding interface shortcuts for frequently executed sequences of commands (e.g., [5,24,27]). As is the case with our work, Debevc *et al.* [7] focus on helping users customize existing interface features. However, their adaptive suggestions are based on a combination of recency and frequency of use, whereas MICA performs a more formal and comprehensive assessment of the impact of customization suggestions on performance. Finally, SUPPLE, a fully-fledged adaptive interface, does include a customization facility that allows users to override the adaptive decisions [10], but it appears to be preliminary given the small amount of detail provided on its implementation. To the best of our knowledge, none of the above systems provide the user with the system's rationale. A complementary body of work focuses on helping users understand the available functionality in feature-rich interfaces, as opposed to reducing complexity through interface customization (e.g., [16,19]).

Direct empirical comparisons of specific adaptive and adaptable interfaces [8,17,21] have shown mixed results, motivating a solution that combines aspects of the two approaches. In two of the evaluations [8,21], the adaptable interface was most often found to be superior. The third evaluation, however, reported a more even division between users who preferred the adaptable interface and those who preferred the adaptive interface [17]. Furthermore, studies comparing adaptive interfaces to static alternatives (e.g., [9,12,23]) have also produced mixed results, indicating that adaptive interfaces can be beneficial, but that care

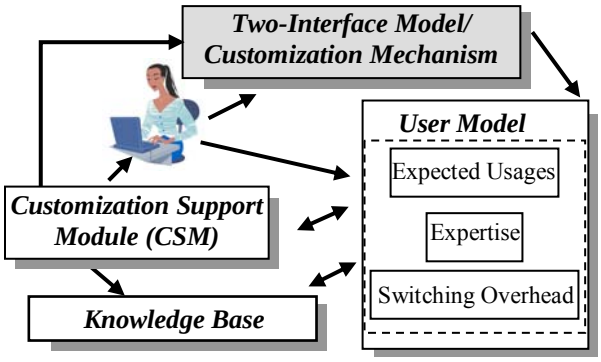

Figure 2: MICA's architecture

should be taken to decide when and how adaptive support should be provided.

We are aware of only one other direct empirical comparison of a mixed-initiative system for GUI customization to either an adaptive or adaptable alternative: the evaluation by Debevc *et al.* [7]. A key difference between our evaluation and theirs is that our evaluation obtains direct information on user interface preference.

SYSTEM FRAMEWORK

MICA's mixed-initiative support is designed to help the user customize given the two-interface model for Microsoft Word (MSWord) proposed by McGrenere *et al.* [21]. The two-interface model, displayed in Figure 1, provides the user with access to two versions of the MSWord interface:

1. **Full Interface (FI)**: the default full MSWord interface (Figure 1, right).
2. **Personal Interface (PI)**: a feature-reduced version of the MSWord interface, containing only features that the user has chosen to add (Figure 1, left).

The motivation for this two-interface model is to allow the user to create a PI that contains only the menu and toolbar items that best suit her needs, but to also allow the user to switch to the FI (using a toggle button) for rarely used features. The PI is built by the user using a lightweight Customization Mechanism: the user enters a mode where the FI is displayed and any feature she selects (as in normal usage) will be added to the PI when she exits the customization mode. A similar mechanism allows users to delete features from the PI. The two-interface model was fully evaluated in a six-week field study [21], ensuring that we are augmenting a customization mechanism that is highly usable.

MICA's mixed-initiative approach relies on finding the user's optimal PI based on the time it would take the user to invoke the features that she needs given the distribution of these features between the PI and FI. Figure 2 depicts

MICA's architecture. The user customizes the PI using the Customization Mechanism described above (a direct extension of the mechanism proposed by McGrenere *et al.* [21]). The Customization Support Module (CSM) is responsible for determining the optimal PI and using it to generate customization suggestions. Determining this optimal interface is done with the help of the User Model, which assesses the user's performance given a particular PI. This performance assessment is based on GOMS analysis [3], a low-cost cognitive modelling technique used to predict the time necessary to perform tasks in a given interface. GOMS analysis has traditionally been used offline to evaluate interfaces and has been shown to be particularly effective at making relative performance comparisons between interfaces [11]. The User Model performs *online* GOMS analysis to evaluate specific customization possibilities. The CSM compares these evaluations and uses the results to make optimal customization suggestions. The relevant GOMS methods are stored in MICA's Knowledge Base. We now describe each component in the framework.

Customization Support Module (CSM)

The CSM decides when to provide the user with tailored customization suggestions and which suggestions to make. To minimize disruption, currently the CSM provides the user with suggestions only when the user initiates customization. These suggestions consist of features that the user should add or remove from her PI and are targeted at optimizing the user's performance with the two-interface model. In general, the more features present in an interface, the greater its complexity, which has the potential to hinder user performance. Therefore, to decide whether to recommend a given set of features for inclusion in the PI, the CSM weighs the extra complexity that these features would introduce into the PI against the time it would take the user to switch to the FI and make the selections from the more complex interface. For an individual feature, this involves a tradeoff between the performance savings that would result from selecting the feature in the PI, versus the *negative* impact this feature's presence in the PI would have on the remainder of the expected PI feature selections. More formally, a feature f_x will be recommended for inclusion in the PI if and only if the following inequality holds, where SelectTime(X, Y) is the time required to perform all expected selections of feature X in interface Y, and EA is the set of all features that are expected to be accessed:

$$SelectTime(f_x, FI) - SelectTime(f_x, PI + f_x) > \sum_{i \in EA - f_x} SelectTime(f_i, PI + f_x) - \sum_{i \in EA - f_x} SelectTime(f_i, PI - f_x) \cdot$$

These selection times depend on i) user-specific information stored in the User Model, ii) the contents of the PI currently under consideration and iii) the layout of both the PI and the FI. To decide which suggestions to make, the CSM performs a greedy search on the space of candidate PIs, each of which has a different subset of features present, to find the one that would maximize the current user's performance. Determining this optimal PI involves asking the User Model to assess the user's expected performance given each candidate PI.

User Model

When the User Model receives a request from the CSM to assess the user's performance for a candidate PI, the User Model (in coordination with the Knowledge Base) estimates how long it will take the user to perform all expected feature selections given that PI. This performance assessment requires the User Model to store information on the following relevant factors (also depicted in Figure 2):

- **Expected Usages:** how often the user is expected to access each feature in the interface. For each feature, the User Model maintains a probability distribution over ranges of plausible access values, with the expected usage defined as the expected value of the distribution.
- **Expertise:** the user's expertise for each feature, where expertise is defined as the amount of time a user takes to locate the feature in the interface. The performance of users with lower expertise will be more negatively impacted by excess functionality than more expert users because it takes lower-expertise users more time to visually search for individual features [1]. For each feature, the User Model represents the expertise of a specific user as a probability distribution over the four expertise categories defined in our previous work [1]: Extreme Expert, Expert, Intermediate and Novice.
- **Switching Overhead:** the amount of time it takes the user to realize that switching to the FI is required for a feature not present in the PI. This allows the User Model to account for the performance implications of having a feature reside solely in the FI if that feature is expected to be used.

To determine appropriate values for the Switching Overhead, we first performed extensive sensitivity analysis to assess the impact of this parameter on CSM suggestions. We then conducted a small study with three participants (ranging from novice to expert users) to obtain a range of plausible values. Since this range lies in a portion of the parameter space where the system is not very sensitive to small deviations, we are currently using the average switching time from the study as an approximation.

By comparison, the sensitivity analysis showed that expected usages and expertise do have a meaningful impact on the CSM's suggestions; however, the User Model does not yet assess them online. There are existing techniques that could readily guide both types of assessments. Expected usages could be assessed through a mixture of plan recognition [2], usage history [13] and dialogues with the user (since MICA is a mixed-initiative system). The Lumiere work [16] could serve as a guide to assessing expertise. We felt, however, that it was first necessary to assess the overall approach before investing time to implement these techniques. Thus, we decided to initially leave expected usages and expertise as black boxes and run a formal user evaluation of MICA's approach, which is described in the "Evaluation" section of this paper.

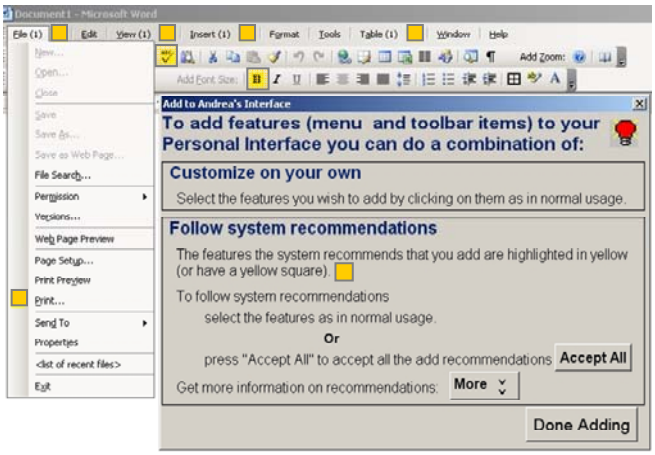

Figure 3: Mixed-initiative customization mechanism

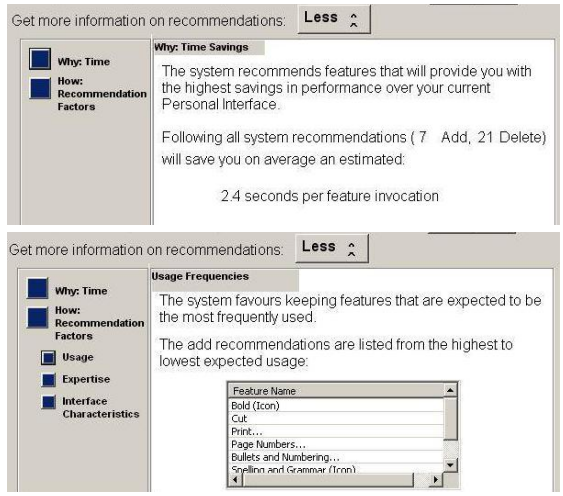

Figure 4: Portions of the system's rationale

Knowledge Base

The Knowledge Base is a GOMS simulation environment whose responsibility is to determine the time necessary for a given user to select a single feature in the two-interface model with a given PI (supplied by the CSM). To predict overall performance, the User Model asks the Knowledge Base for the time necessary to select each feature with an expected usage greater than zero.

The GOMS simulation environment is our extension of the GLEAN tool [18], which, given an interface layout, generates performance predictions of the following basic operations on an interface item: 1) visually searching, 2) pointing and 3) clicking. We extended GLEAN to generate *expected* visual search predictions based on the probabilistic assessment of user expertise generated by the User Model. Thus, for each feature in question, the User Model provides the Knowledge Base with the expertise probability distribution. For each value in the distribution, the extended GLEAN performs a visual search calculation appropriate for that expertise level (see [1] for details). The *expected* visual search time is the expected value of these calculations.

In its original form [3], GOMS predicts performance based on the assumption of a highly skilled user. Our current GLEAN extension accounts for varying levels of expertise in one specific way – namely the impact of expertise on the time necessary to visually search for features (given the number of items present in the interface). Expertise and interface complexity, however, may interact in a number of other ways. For instance, as interface complexity increases, novice users may make more feature-selection errors than experienced users. Extending our GOMS simulation environment to account for such additional impacts of user expertise is an area for future investigation.

DELIVERING THE ADAPTIVE SUPPORT

To avoid some of the common disadvantages of purely adaptive interfaces, the delivery of MICA's customization suggestions is designed to 1) maintain user control, 2) provide customization support non-intrusively, and 3) maintain predictability and transparency.

As mentioned above, MICA provides customization recommendations only when the user initiates customization (i.e., clicks the "Modify" button in Figure 1). Figure 3 shows MICA's mixed-initiative customization interface for adding features. When the user enters this mode, the FI is displayed, along with the dialogue box located in the central part of Figure 3. MICA's recommended additions are made visually distinct by highlighting (in yellow) recommended toolbar items (see top of Figure 3) or by having squares (also yellow) beside recommended menu items (see the pull-down menu in Figure 3) and beside menu headings with recommended features inside them. Our original intention was to highlight the entire menu item/heading in yellow, but this was not possible with the available API for MSWord. Users can accept the recommendations by selecting the features as in normal usage. Alternatively, the "Accept All" button in the dialogue box allows users who trust the system's recommendations to accept all of them at once. Users maintain control because it is ultimately up to them to decide when and how to customize and to what degree they wish to follow MICA's recommendations. The design of the mixed-initiative interface was informed by informal usability testing with pilot participants.

To maintain predictability and transparency [14], the user is provided with access to MICA's rationale. This includes a description of *why* MICA is making recommendations and *how* it generated them. This explanation can be obtained by clicking the "More" button in Figure 3, which expands the dialogue box to include an additional pane of information. The *why* component of the rationale, which is shown at the top of Figure 4, explains that MICA is making recommendations because it predicts the recommendations will save the user time. MICA reports the average estimated time savings per feature invocation should the user choose to accept all recommendations (i.e., the entire optimal PI). The *how* component of the rationale describes three factors that impact the system's decisions: 1) usage frequencies, 2) expertise and 3) interface characteristics. The user can obtain more information on each factor, which consists of an explanation of this factor and, if relevant, access to a high-level snapshot of the User Model's assessment. Figure 4

displays the extra information for the "Usage Frequencies" factor. The rationale displayed in Figure 4 represents our first attempt at conveying this information to the user and we expect that its design will have to be refined through user evaluations. While explaining adaptive behaviour has been explored in other contexts (e.g., [6]), this is the first attempt to show system rationale in GUI customization research. As a result, evaluation is necessary to ascertain what types of information, if any, users find useful, along with how to convey the information.

EVALUATION

The goal of the evaluation was to gain an initial understanding of the value of MICA's mixed-initiative approach. We felt that the most appropriate first evaluation of the system would be to run a lab study comparing the mixed-initiative interface to an adaptable alternative. Therefore, we conducted an experiment with two conditions: 1) the purely adaptable two-interface model, where users could customize but did not receive system recommendations, and 2) MICA's mixed-initiative interface described above. The conditions were identical except for the mixed-initiative component. We chose this overall design to provide insight into a number of issues. 1) Do users prefer the mixed-initiative support to customizing on their own? 2) Does MICA's support have positive effects on task performance? 3) How does the presence of recommendations impact customization behaviour?

Design

The experiment used a within-subjects factorial design with interface type (Mixed-Initiative or Adaptable) as the primary factor. Participants completed two tasks, one with each version of the interface (described in the "Tasks" section of this paper). Therefore, task was a within-subjects control variable. Both interface order and task order were between-subject controls. To account for learning effects, we counterbalanced the order of interface and tasks, resulting in four configurations.

A within-subjects design (i.e., each participant completes both conditions) was chosen to gain direct preference data and to account for variability owing to individual differences. In addition, this design requires fewer participants, since each participant provides data for two conditions.

Participants

Twelve participants completed the study (nine females, three males). Participants were recruited by posting signs around the university campus. All participants were in the 18-29 age range, except one participant who was 50-59. Ten were students, one was an admin manager, and one was a retired teacher. Participants were paid $10/hr.

Prior to signing up for the experiment, interested participants first completed a preliminary questionnaire developed by McGrenere and Moore [22] that classifies users as either Feature Keen, Feature Shy or Feature Neutral based on their answers to questions on the following: i) how they feel about having many functions in the interface, ii) how much they want to have a complete version of their interface and iii) how up-to-date they would like their interface to be. We selected only Feature Keen and Feature Shy participants (an equal number of each) because we wanted to avoid having a large number of participants who may have little opinion on their interfaces (i.e., Feature Neutrals). While we did want our participants to care about the state of their interfaces, we did not anticipate Feature Keen vs. Feature Shy differences with respect to our independent variable (mixed-initiative vs. adaptable).

Apparatus

The experiment was conducted on an IBM Thinkpad running Windows XP with a 2.0 GHZ processor, 1.5 MB RAM, and a 15" screen. The adaptable and mixed-initiative interfaces were coded for MSWord 2003 using Visual Basic for Applications (VBA) macros. The mixed-initiative framework was implemented in C++.

Tasks

One of the biggest challenges in designing a lab study involving customization is how to motivate users to customize, since customization is typically meant to be beneficial over a period of time longer than a lab study. Thus, the experimental tasks had to be designed such that: 1) customization would *actually* have the potential to be beneficial; and 2) participants would *feel* that customization could be beneficial. Satisfying both constraints required that: i) the tasks be designed so that participants would spend most of the task time selecting features from the menus and toolbars as opposed to entering text; and ii) participants would feel that there was enough regularity in the feature usage to be worth customizing their interfaces.

Participants performed two tasks (A and B) similar in length and overall complexity, one with each version of the interface. Each task consisted of a list of step-by-step instructions and a target final document. Each step described an interface operation to be performed (or in some cases, a small amount of text entry) and indicated whether to use the toolbars or menus to complete the step, but did not explicitly give the name of the command. We refer to this type of task as a *guided task*. The restricted nature of the guided tasks served two purposes: 1) the tasks required a large number of menu selections and could still be completed within a reasonable-length session (3 hours), and 2) we were able to assign accurate values to the expected usage component in the User Model (described below).

Alternatives to guided tasks include asking participants to select a stream of named menu and toolbar features (an approach used in a previous study comparing an adaptive and adaptable interface [8]), or a more open-ended task, such as "write a short report on topic X." We wanted to make our tasks somewhat more ecologically valid and engaging than the selection stream alternative, however, an open-ended task would result in too few menu selections in the same study duration and less accurate information for the User Model. Guided tasks appeared to strike the right balance between these two extremes.

To further motivate customization, we used task repetition

in combination with a small amount of deception. Each task was actually repeated three times; however, to make customization appear even more beneficial, participants were told that each task would be repeated up to five times. The customization mechanism was enabled only after the first repetition of each task to allow participants to become familiar with the task before customizing. To motivate usage of the PI, for each task, participants were given a starting PI that contained many, but not all of the features required for the task and some features that were not needed.

Procedure

The procedure for the experiment was as follows. 1) Participants completed a detailed questionnaire designed to assess their expertise for each feature used in the experiment. 2) The questionnaire results and the information on each feature's anticipated usage frequency in each of the guided tasks were used to initialize the User Model. Recall that this is necessary because the User Model currently cannot assess these online. 3) The two-interface model and customization mechanism were briefly demonstrated to the participants using the interface (i.e., mixed-initiative or adaptable) to be present during the first trial. 4) Participants performed the first guided task (repeated three times, the customization mechanism was enabled after the first of three repetitions). 5) The interface to be used in the second trial was introduced. For the mixed-initiative interface, this new interface was briefly demonstrated. For the adaptable interface, participants were simply told that the customization mechanism would no longer contain system recommendations. 6) Participants performed the second guided task (repeated three times, the customization mechanism was enabled after the first of three repetitions). 7) Participants completed a post questionnaire. 8) Participants were interviewed by the first author. A session typically lasted 2 hours and 30 minutes, but ranged from 2 to 3 hours.

Measures

Our evaluation had a number of quantitative and qualitative measures. These measures are described below, grouped according to category.

Performance:

- Overall Performance: the amount of time it took the participants to complete the tasks overall, including customization time.
- Task Performance: the amount of time the participants spent on the tasks, ignoring time spent in the customization mechanism.

Customization Behaviour:

- Customization Time: the amount of time spent in the customization mechanism when some customization actually occurred.
- Features Added/Deleted: the total number of features added/deleted to/from the Personal interface.

Impact of Recommendations on Customization Decisions: In addition to the above quantitative within-subjects measures, we also measured how user customizations matched system recommendations in the mixed-initiative condition. In the adaptable condition we measured how user customizations matched what the system would have recommended.

Interface Preference: On the post questionnaire participants were asked to state which of the two interfaces they would install on their machine (Overall Preference). Participants were also asked to state which interface they preferred (or if they found them equal) for the following five criteria: 1) the ease of use (Easy of Use), 2) the ease of adding features to the PI (Easy to Add), 3) the ease of deleting features from the PI (Easy to Delete), 4) whether the PI matched their needs after customization (Match Needs), and 5) the time necessary to customize (Fast).

Reasons for Customizing and *Feelings Towards Recommendations:* The post questionnaire also asked participants to rate (on a five-point scale) three potential reasons for customizing (listed in Table 2) and their feelings towards the system recommendations on four dimensions (listed in Table 3). Additional feedback on customization, the recommendations and the rationale were gathered in the interview.

Results

Out of the 12 participants who completed the study, 8 customized in both conditions. The remaining 4 participants customized only in one condition: 3 in the adaptable condition and 1 in the mixed-initiative condition. For 3 of these participants, the customization occurred in the second condition. Unless otherwise specified, the results presented in this section are based on the data from only the 8 participants who customized in both conditions.

The within-subjects quantitative dependent measures were analyzed using univariate ANOVA with *interface* (mixed-initiative or adaptable) as the primary within-subjects factor. Two between-subjects control factors were included in the analysis as a result of the counterbalancing*: Interface Order* and *Task Order*. Along with statistical significance, we report partial eta-squared (η^2), a measure of effect size. Effect size measures the practical significance of the differences found. To interpret this value, .01 is a small effect size, .06 is medium, and .14 is large [4].

Before performing the ANOVA analysis with *interface* as the primary within-subjects factor, we checked for an effect of task (A vs. B). As expected, we did not find a main effect of task on any of our dependent measures. We begin by describing how *interface* impacted performance and customization behaviour (summarized in Table 1).

Performance: For Overall Performance, participants were faster in the mixed-initiative condition, spending an average of 28 minutes 6 seconds total time compared to 30 minutes 19 seconds in the adaptable condition. The analysis revealed a marginally significant main effect of *interface* for this dependent measure ($F(1, 4) = 6.522$, $p = 0.063$) with a large effect size (partial $\eta^2 = 0.620$). The results were similar when considering Task Performance only (see Table 1).

Dependent Variable	Mean (SD) *MI*	Mean (SD) *AD*	F(1,4)	p	η^2
Overall Performance (minutes)	28:06 (6:09)	30:19 (5:29)	6.522	0.063	0.620
Task Performance (minutes)	26:40 (5:29)	28:44 (5:05)	6.587	0.062	0.622
Customization Time (minutes)	1:06 (0:33)	1:35 (0:38)	8.170	0.046	0.671
Features Added	6.1 (0.8)	6.8 (1.5)	2.778	0.171	0.410

Table 1: Results for the quantitative within-subjects measures (N = 8) MI=Mixed-Initiative, AD = Adaptable

While both results are only marginally significant, the fact that they had large effect sizes implies that the mixed-initiative had a large impact on both performance measures.

Customization Behaviour: We analyzed the impact of *interface* on customization behaviour in terms of both the time necessary to customize and the number of features that participants added and deleted. For Customization Time, Table 1 shows that participants spent significantly less time customizing in the mixed-initiative condition than in the adaptable condition ($F(1,4) = 8.170$, $p=0.046$, partial $\eta^2 = 0.671$).

There was also a significant between-subjects main effect of *Interface Order* ($F(1,4)= 10.062$, $p=0.034$, partial $\eta^2 = 0.716$), showing that participants spent less time customizing across both conditions if they saw the adaptable condition first (average: 55 seconds, SD: 24.4 seconds) than if they saw the mixed-initiative interface was first (average: 1 minute 46 seconds, SD: 12.3 seconds). Interpreting this order effect is difficult since there were only 4 participants per order and there appeared to be large individual differences. This may be an indication, however, that the simpler adaptable interface provided scaffolding for the mixed-initiative interface and not vice versa. This result could also potentially be attributed to the fact that participants in the Adaptable/Mixed-Initiative order received an additional interface demonstration (see the "Procedure" subsection).

While *interface* did have an effect on Customization Time, it did not appear to impact the number of features added, since the difference between the two conditions was not significant ($F(1, 4) = 2.778$, $p=0.171$). The fact that participants added roughly the same number of features in both conditions implies that the faster customization time with the mixed-initiative interface was not caused by participants failing to customize.

None of the 12 participants deleted any features. When asked why in the interview, the majority of participants (8) said that the extra features in the PI weren't bothersome (67%). Four explicitly mentioned that it wasn't worth the time necessary to delete them (33%); 2 said that the PI was small enough already (17%); and 2 thought the extra features might be useful at some point (17%). The percentages

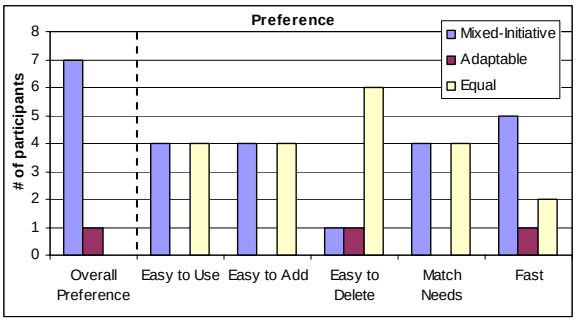

Figure 5: Preference rankings for the two interfaces (N=8)

do not sum to 100 because some users provided more than one reason.

When analyzing the data for Customization Time, we discovered that the design of the "Accept All" feature (shown in Figure 3) may need to be revisited. This more automatic form of customization was utilized by five participants for some portion of their customization. After customizing using "Accept All", one participant had particular difficulty remembering what she had customized, entering the mechanism an additional seven times without customizing. In the interview, this participant revealed that she would enter the customization facility when she couldn't immediately locate the feature in the PI.

Impact of Recommendations on Customization Decisions: We found that the 9 participants in the mixed-initiative condition followed the vast majority of the system's recommended additions. Specifically, out of the total number of "add" recommendations, 96% were followed. Participants in the mixed-initiative condition also added features that were not recommended. In particular, 11% of their customizations were not recommended features.

To gain an understanding whether participants would have made the same customization decisions with or without the system's help, we also examined the customizations of the 11 participants in the adaptable interface. We compared their customizations to what would have been recommended by the mixed-initiative system (with the User Model appropriately initialized) and found that these participants added only 78% of features that would have been recommended (compared to the 96% discussed above). Furthermore, 35% of their customizations did not match features that would have been recommended (compared to the 11% discussed above).

The above results in conjunction with the performance results indicate that the recommendations impacted the participants' customization decisions in a positive manner. The data provides encouraging evidence that in the mixed-initiative condition, participants followed the recommendations, added fewer non-recommended features and performed better than those in the adaptable condition.

Interface Preference: In addition to the above mainly quan-

Reasons for Customizing	Mean	SD
To reduce the number of features that had to be accessed using the Full Interface.	3.09	1.70
To make the Personal Interface as small as possible while still being appropriate for the tasks.	3.45	1.39
To help complete the tasks more quickly.	4.64	0.50

Table 2: Reasons for customizing ranked on a 5-point scale (N=12)

titative results, our evaluation also provides qualitative support for the mixed-initiative interface. For example, our within-subjects design allowed us to obtain direct preference information. Figure 5 displays these results. For Overall Preference, MICA's mixed-initiative interface was preferred by 7 of the 8 participants who customized in both conditions. This is consistent with the individual criteria, which showed that participants either preferred the mixed-initiative interface or found the two to be equal on all criteria. The two exceptions to this are that one user rated adaptable best for Easy to Delete and another user rated adaptable best for Fast. For Easy to Delete, all responses were hypothetical since none of the participants entered this mode of customization. The participant who found customization faster with the adaptable interface (and preferred it overall) was an expert user who said that he knew exactly which features to add and found that there was too much text in the dialogue box describing the customization procedure in the mixed-initiative interface (Figure 3).

Reasons for Customizing: Table 2 summarizes the participants' responses (for all 12 participants) concerning the three potential reasons for customizing. The fact that the highest rated reason is task performance is encouraging, since it forms the basis of MICA's recommendations. A free-form section provided participants with an opportunity to list additional reasons for customizing. Three participants entered comments in this section, indicating that the nature of the experimental setup was also a factor (e.g., the restricted/repetitive nature of the tasks and the novelty of the interface). During the follow-up interviews, participants were also asked why they customized and why they added features. Again, performance had the most support (50%) but some also liked the simplicity of the PI (34%) and some seemed to want to use it exclusively (17%).

Feelings Towards Recommendations: On the post-questionnaire, the 9 participants who customized in the mixed-initiative condition were asked to rank aspects of the system's recommendation. Table 3 summarizes these results. Overall responses were positive, as were responses on a free-form section of the questionnaire that asked participants to indicate what they liked about the recommendations. Most participants stated that they liked how the recommendations were presented (33%) or that the recommendations were appropriate for their tasks (55%).

In terms of what participants disliked about the recommendations, some wanted all features needed for the task to be recommended (22%). Three participants pointed to three different usability issues: 1) having to look through the menus to see what was recommended, 2) the amount of text in the dialogue box, and 3) the fact that the entire menu heading wasn't highlighted (see Figure 3).

Statement About Recommendations	Mean	SD
I trusted the system to make good recommendations	4.11	0.60
It was easy to tell which features were recommended.	3.78	1.09
Recommendations were appropriate for the tasks.	4.44	0.53
I understood why the system made the recommendations that it did.	4.00	1.00

Table 3: Feelings towards recommendations ranked on a 5-point scale (N = 9)

Rationale: Despite not being an explicit focus of the study, we had hoped that at least some users would view MICA's rationale in order to provide preliminary information on its usefulness. Unfortunately, none of the participants in the study chose to look at the rationale. In the post-session interview, the majority of participants indicated that they either were too focused on completing the tasks to look at it (44%) or didn't need the information (44%). It is important to note, however, that participants were not aware of what information was in the rationale, since this feature of the mixed-initiative interface was not demonstrated during the interface training.

Feature Keen/Shy Differences: While we did not anticipate the Feature Keen/Shy classification to impact our main quantitative or qualitative measures, we did look for any clear trends. One noticeable difference was that out of the 8 participants that customized in both conditions, 6 were Feature Keen and 2 were Feature Shy. In addition, Feature-Shy participants rated all the customization reasons listed in Table 2 higher (on average) than the Feature-Keen participants. These findings may suggest a difference between these two groups with respect to customization, although not necessarily in terms of mixed-initiative interfaces. Further research would be required to substantiate this, before which additional validation work on the Feature Shy/Keen questionnaire may be needed.

Discussion

Our results provide encouraging evidence that when MICA has a fairly accurate User Model, users prefer the mixed-initiative system to the purely adaptable alternative. In addition to users preferring MICA's support, participants followed the vast majority of the system's recommended additions (96%) and the data suggests that these recommendations helped improve performance in terms of time on task. Although the performance differences are small, this is to be expected given the relatively short duration of the study. The time savings should add up given longer periods of use in real settings, especially if the user's tasks maintain a certain amount of consistency. Exactly how much consistency must be present in the user's feature usage to make customization beneficial, in terms of both objective and sub-

jective benefit, remains an area of future investigation. Feature usage doesn't necessarily have to be as restricted as it was our study. However, if the set of regularly used features is changing dramatically and frequently, then the time to customize may begin to outweigh any benefit.

The data also shows that MICA's support has the potential to decrease customization time. Since the effort necessary to customize is one of the disadvantages of purely adaptable interfaces, decreasing customization time may make users more willing to customize. The results, however, also point out a potential downside of allowing the system to do more of the customization on behalf of the user. For one user in particular, the more automatic form of customization led to her having difficultly remembering what she had already added to her PI when it came to features that she was not as familiar with. However, not all participants who used this feature experienced these problems, indicating that this more automatic form of customization may be problematic for some and not others. Furthermore, it may simply be a matter of the current confirmation dialogue, which displays the names of the newly added and deleted features, being insufficient for some users.

While the evaluation provides support for MICA basing its recommendations on performance and for the general appropriateness of its recommendations, the evaluation also points to two assumptions embedded in MICA decision-making process that require further exploration. First, MICA assumes that users will be willing to switch to the FI for less-frequently used features, however, 22% of users indicated that having features "missing" from the recommendations was something they disliked about the system. It would be interesting to explore whether some users would prefer to solely use the PI, regardless of the performance impact, or whether better understanding this performance tradeoff would influence their preference. Second, MICA assumes users will be willing to delete features, which in the context of our study, was not the case. Users where also reluctant to delete features in McGrenere *et al.*'s field study [21].

Finally, a number of study limitations deserve mention. First, since it was a lab study, we do not know how users would respond to the mixed-initiative support if they were using the system on a day-to-day basis, performing tasks that have less obvious structure. McGrenere *et al.*'s field study [21] does provide evidence that most users saw enough regularity in their feature usage in a real working context to believe that customization would be beneficial. Thus, there is reason to believe that users would see the value of mixed-initiative support in such contexts as well. Second, it remains to be seen whether or not our results will hold in either the lab or the field when the User Model is performing online assessment of user expertise and expected feature usage, which will be less accurate than the settings we have used in our study. Third, our evaluation indicates that the system recommendations had positive influence on the user customization decisions; however, it did not allow us to directly test the impact of all user-model and decision-making parameters. Despite the above limitations, we believe that our evaluation is an important first step. Showing that this type of mixed-initiative support can be beneficial motivates investigating appropriate online assessment techniques, in addition to conducting more detailed evaluations, first in the lab and then in the field.

CONCLUSIONS AND FUTURE WORK

In this paper we described MICA, a mixed-initiative framework that provides interface customization suggestions tailored to the user's work patterns, expertise and characteristics of the interface itself. MICA performs online GOMS analysis, combining this user- and interface-specific information to generate suggestions that improve task performance. Explicit focus on performance is unique to our system. Other work on interface customization assumes that customization saves time but never formally quantifies these savings to make more informed decisions on how to customize. To avoid disrupting the user, MICA presents its recommendations only when the user initiates customization and provides them in a non-intrusive manner. To ensure that the recommendations are predictable and transparent, MICA provides the user with access to its rationale.

In a formal evaluation, we compared MICA's mixed-initiative interface to a purely adaptable alternative. The results indicate that users prefer the mixed-initiative support, at least in circumstances where the system has an accurate user model. The evaluation also provides encouraging evidence that MICA's recommendations improve time on task and decrease customization time. Furthermore, the evaluation is one of the few direct comparisons of a mixed-initiative and adaptable interface, and thus it extends the body of knowledge pertaining to the value of mixed-initiative approaches. It is particularly encouraging that the mixed-initiative support was preferred to an adaptable interface that had performed well in a previous study.

Since our evaluation did not provide any insight into the value of showing the user the rationale for the system's customization suggestions, we are currently running a follow-up evaluation designed to test the effects of viewing the rationale on user customization behaviour. Additional areas of future work involve extensions to MICA's framework. First, we would like to extend the framework to allow MICA to make recommendations without waiting for the user to initiate customization, which could be beneficial for users who are forgetting to customize. Adding this functionality will involve weighing the expected performance improvement against the cost of *interrupting* the user to make the suggestions. A second potential extension would be to reason about *when* a feature is expected to be used in addition to how often, both in deciding which recommendations to make and how to deliver them to the user. We would also like to explore the generalizability of our approach. While our current implementation of MICA's framework is for MSWord, the underlying principles generalize to any menu/toolbar interface. Thus, it would be interesting to apply this approach to other classes of appli-

cations with different levels of complexity. Finally, we plan to investigate appropriate online assessment techniques for the User Model.

ACKNOWLEDGEMENTS

We thank Kasia Muldner and Steph Durocher for commenting on previous drafts of this paper. We also thank the anonymous reviewers for their comments and suggestions.

REFERENCES

1. Bunt, A., Conati, C., and McGrenere, J. What Role Can Adaptive Support Play in an Adaptable System? In *Proc. of IUI*, 2004, pp. 117-124.
2. Carberry, S. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, *11*, 1-2 (2001), 31-48.
3. Card, S. K., Newell, A., and Moran, T. P. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ 1983.
4. Cohen, J. Eta-squared and partial eta-squared in communication science. *Human Communication Research*, *28*, (1973), 473-490.
5. Cypher, A. EAGER: Programming Repetitive Tasks by Example. In *Proc. of CHI*, 1991, pp. 33-39.
6. Czarkowski, M. and Kay, J. How to Give the User a Sense of Control Over the Personalization of Adaptive Hypertext? In *Proc. of Adaptive Hypermedia and Adaptive Web-Based Systems (in conjunction with UM'03)*, 2003, pp. 121-131.
7. Debevc, M., Meyer, B., Donlagic, D., and Svecko, R. Design and evaluation of an adaptive icon toolbar. *User Modeling and User-Adapted Interaction*, *6*, 1 (1996), 1-21.
8. Findlater, L. and McGrenere, J. A Comparison of Static, Adaptive, and Adaptable Menus. In *Proc. of CHI*, 2004, pp. 89-96.
9. Gajos, K., Czerwinski, M., Tan, D. S., and Weld, D. S. Exploring the Design Space for Adaptive Graphical User Interfaces. In *Proc of AVI*, 2006, pp. 201-208.
10. Gajos, K., D. Christianson, R. Hoffmann, T. Shaked, Henning, K., Long, J. J., and Weld, D. S. Fast and Robust Interface Generation for Ubiquitous Applications. In *Proc. of Ubicomp*, 2005, pp. 37-55.
11. Gong, R. and Kieras, D. A Validation of the GOMS Model Methodology in the Development of a Specialized, Commercial Software Application. In *Proc. of CHI*, 1994, pp. 351-357.
12. Greenberg, S. and Witten, I. H. Adaptive personalized interfaces - a question of viability. *Behaviour and Information Technology*, *4*, 1 (1985), 31-45.
13. Greenberg, S. and Witten, I. H. How Users Repeat Their Actions on Computers: Principles for Design of History Mechanisms. In *Proc. of CHI*, 1988, pp. 171-178.
14. Hook, K. Steps to take before intelligent user interfaces become real. *Interacting with Computers*, *12*, (2000), 409-426.
15. Horvitz, E. Principles of Mixed-Initiative User Interfaces. In *Proc. of CHI*, 1999, pp. 159-166.
16. Horvitz, E., Herckerman, D., Hovel, D., and Rommelse, R. The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In *Proc. of UAI*, 1998, pp. 256-265.
17. Jameson, A. and Schwarzkopf, E. Pros and Cons of Controllability: An Empirical Study. In *Proc. of AH*, 2002, pp. 193-202.
18. Kieras, D. E., Wood, S. D., Abotel, K., and Hornof, A. J. GLEAN: A Computer-Based Tool for Rapid GOMS Model Usability Evaluation of User Interface Designs. In *Proc. of UIST*, 1995, pp. 91-100.
19. Linton, F. and Schaefer, H. Recommender systems for learning: building user and expert models through long-term observation of application use. *User Modeling and User-Adapted Interaction*, *10*, 2-3 (2000), 181-207.
20. Mackay, W. E. Triggers and Barriers to Customizing Software. In *Proc. of CHI*, 1991, pp. 153-160.
21. McGrenere, J., Baecker, R. M., and Booth, K. S. An Evaluation of a Multiple Interface Design Solution for Bloated Software. In *Proc. of CHI*, 2002, pp. 163-170.
22. McGrenere, J. and Moore, G. Are We All in the Same "Bloat"? In *Proc. of GI*, 2000, pp. 187-196.
23. Mitchell, J. and Shneiderman, B. Dynamic versus static menus: an exploratory comparison. *SIGCHI Bull.*, *20*, 4 (1989), 33-37.
24. Oppermann, R. Adaptively supported adaptability. *International Journal of Human-Computer Studies*, *40*, (1994), 455-472.
25. Shneiderman, B. and Maes, P. Direct manipulation vs. interface agents. *interactions*, *4*, 6 (1997), 42-61.
26. St. Amant, R. and Cohen, P. R. Interaction with a Mixed-Initiative System for Exploratory Data Analysis. In *Proc. of IUI*, 1997, pp. 15-22.
27. Thomas, C. G. and Krogsoeter, M. An Adaptive Environment for the User Interface of Excel. In *Proc. of IUI*, 1993, pp. 123-130.