Triggering Triggers and Burying Barriers to Customizing Software

Nikola Banovic[1,2], Fanny Chevalier[3], Tovi Grossman[1], and George Fitzmaurice[1]

[1]Autodesk Research
210 King St. East, Toronto,
Ontario, Canada, M5A 1J7
{firstname.lastname}@autodesk.com

[2]Dept. of Computer Science
University of Toronto, Toronto,
Ontario, Canada, M5S 3G4
nikola@dgp.toronto.edu

[3]OCAD University
Toronto, Ontario, Canada, M5T 1W1
fchevalier@ocad.ca

ABSTRACT

General-purpose software applications are usually not tailored for a specific user with specific tasks, strategies or preferences. In order to achieve optimal performance with such applications, users typically need to transition to an alternative efficient behavior. Often, features of such alternative behaviors are not initially accessible and first need to be customized. However, few research works formally study and empirically measure what drives a user to customize. In this paper, we describe the challenges involved in empirically studying customization behaviors, and propose a methodology for formally measuring the impact of potential customization factors. We then demonstrate this methodology by studying the impact of different customization factors on customization behaviors. Our results show that increasing exposure and awareness of customization features, and adding social influence can significantly affect the user's customization behavior.

Author Keywords

Adaptable interfaces; adaptive interfaces; customization; mixed-initiative; personalization.

ACM Classification Keywords

H.5.2. [Information interfaces and presentation]: User Interfaces – Graphical user interfaces.

General Terms

Human Factors, Measurement, Performance.

INTRODUCTION

One of the aims of HCI research is to design interfaces that allow users to maximize their performance while interacting with the computer. However, for complex software applications, supporting optimal efficiency while remaining universally accessible is a challenge [33]. For example, many desktop applications are designed to accommodate a broad spectrum of users and usage scenarios. These general-purpose interfaces are usually not tailored for a specific user with specific tasks, strategies or preferences. As such, the most apparent way to use software is rarely the most efficient way for any particular user. Users typically need to adapt their own behaviors if they are ever to achieve optimal performance.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

In some cases, alternative enhanced behaviors are, by default, available for use. An example is preconfigured hotkeys. However, in other cases, features that would lead a user to more optimal performance are not initially accessible and first need to be customized [18].

While evidence suggests users spend a considerable amount of time personalizing the appearance of their interfaces [23] (*i.e.* customizing aesthetics), they can be reluctant to customize the functionalities to perform their tasks more efficiently [21] (*i.e.* customizing behaviors). In particular, users are reluctant to change their behavior when they are in the middle of a production process [4]. However, at some *breaking point*, a user may decide to perform a customization. This may be triggered when the perceived benefits outweigh the costs of setting up the customization. If software systems can manipulate this *breaking point*, then users could more rapidly approach optimal performance.

Although many studies look at the performance of customizable interfaces, few formally study and empirically measure what drives a user to customize. Typically, customizable interfaces are evaluated on the assumption that the user *does* choose to customize [9]. Customization factors established by Mackay [21] are probably the most thorough to date, but the identified "triggers" and "barriers" have not yet been empirically measured. In particular, it is worthwhile to explore whether such factors can be manipulated through the software application itself, to increase the customization behavior of its users, and ultimately, optimize their performance.

With these goals in mind, we present our contributions, which are both methodological and empirical. First, we describe the challenges involved in empirically studying customization behaviors, and propose a methodology for formally measuring the impact of potential customization factors. We then demonstrate this methodology by studying the impact which exposure, awareness, and social factors have on customization behaviors. Our results show that users develop different customization strategies, and that some might be more eager to customize and switch to an *enhanced behavior*. But more importantly, we also show that increasing exposure and awareness, or adding social influences, can significantly impact a user's *breaking point*. This implies that there exist customization factors which can be manipulated in software.

RELATED WORK

Our work is related to research on novice to expert transitions, personalized user interfaces, customization facilities, and customization behaviors.

Learning and Novice-Expert Transition

Grossman *et al.* [17] explicitly identified the lack of transitioning to a more expert approach as one of five main learnability problems in software. Past research looked at ways to help the user make this transition by explicitly teaching how to use an alternative approach [18, 25, 31], or even forcing the user to use the more efficient approach [16]. That body of work, however, does not focus on the customization portion of the transition.

Personalized User Interfaces

Software personalization can improve task performance and reduce workload in GUI control structures. We refer the reader to Findlater and McGrenere [12] for a detailed overview of such user interfaces.

Prior work has proposed numerous adaptive user interfaces that can improve user performance compared to static interfaces. Most adaptive approaches looked at ways to improve the performance of accessing items from a menu [6, 11, 13, 15, 29, 35], with variations on frequency-based menus [27] and split menus [14, 32] being very common. Another example in the research literature is layered interfaces [5, 10, 26, 33] which provide multiple levels of interface complexity, to match a user's skill level. Some of these adaptable interfaces also showed benefits over adaptive ones, with users performing faster [9], or showing preference for adaptable interface over adaptive ones [26].

Typically, when such techniques are evaluated, the main goal was to study the benefits of customization under the assumption that the user customized, and not whether or not the user would *choose* to customize [9]. Therefore, it still remains important to formally study what causes a user to choose to customize.

Customization Facilities

McGrenere *et al.* [26] pointed out that for some users the amount of time it took to adapt the system was an inhibitor to customization. As such, the *customization facility* should be efficient. Research has proposed to simplify customization by providing in-place customization features (*e.g.* from user's command history [19], or by selecting any GUI widget in-place and replicating it onto a command palette [34]). Neither of these two customization facilities has been formally evaluated.

Mixed-initiative incremental interfaces [1, 36], and multiple-interfaces [2] could improve the user's performance. Such interfaces provide mechanisms to prompt the users to make a customization based on user's behavior, proficiency, or current task. An example is the Adaptive Bar [7], which prompts the user to add a command to an application bar based on historical frequency of the command. The evaluation of that system showed advantage over a purely adaptable system, and also noted that novice users mostly did not customize the purely adaptable system.

Customization Behavior

Users often customize software because they wish to personalize it, improve their performance, or reduce their workload while using the software [23]. However, Mackay [21] showed that most users do not customize, and identified a comprehensive list of triggers as well as barriers to customization. However, few studies have empirically evaluated the impact of these factors on customization behaviors.

Past research has investigated user behaviors with different personalized systems. Marathe and Sundar [24] showed that customization behavior could depend on the proficiency of the user, and that power-users might be more inclined to customize to gain a sense of control over the software.

Social influence has also been identified as an important factor that could influence user customization behavior [20, 21, 37] and could encourage transition to expert use [30]. Research has shown that often there exist users within organizations that customize software and are willing to share their customizations [22], which could also be beneficial as it prevents intelligibility problems associated with local customization [8]. In our study we investigate how we can encapsulate this social influence into a mixed-initiative system in order to influence customization.

CUSTOMIZATION FACTORS

We define a user's *breaking point* as when a user decides to customize their software. The factors which Mackay [21] identified are all potential influences of a user's *breaking point*. Those factors describe the contexts and the social processes in which the user is more likely to customize. If software systems could influence this *breaking point* by manipulating those customization factors, then users could more rapidly approach optimal performance.

However, some of the customization factors listed by Mackay [21] are beyond the control of the software application (*e.g.* external factors such as job changes). Other customization factors can potentially be manipulated, which we discuss below. There, we draw parallels between these customization factors and factors that drive technology adoption in general. We refer to the factors in Unified Theory of Acceptance and Use of Technology (UTAUT) [38]: performance expectancy, effort expectancy, social influence, and facilitating conditions.

Awareness. One of the first important triggers of customization is the user's performance expectancy, or realization that there is an alternative way of performing the task, which improves the user's performance or reduces the user's workload. In order to customize software, the user must also be aware of the facilitating conditions, or the customization facility. If software could increase awareness of the mechanism for performing customizations, it could influence the user's *breaking point*.

Exposure. It is often the case that the user is aware that customization is possible, but believes that the process is too complicated. The inability of the user to determine the actual time it would take to switch from one way of performing a task to another, where the time includes the time to customize, could be a barrier to customization. On the other hand, exposing the user to the customization could change user's effort expectancy and trigger a customization.

Social influences. The social influence to perform better or simply like other users could be a strong customization trigger. Furthermore, the recommendation from a friend to customize could influence a user's perceived benefit of the customization. Integrating social cues with a software environment may allow for such triggers.

METHODOLOGY CONSIDERATIONS

As we have discussed, software customization takes place over time and is influenced by many factors. It is thus worthwhile to empirically evaluate the relative impact of these factors. Our goal is to develop a methodology where during the study, the user can freely choose between using some *default behavior* (*e.g.* using a menu), or choose to customize using a *customization facility* (*e.g.* a customization dialog box), to enable an *enhanced behavior* (*e.g.* using a hotkey). The primary dependent variable of the methodology is if and when a user chooses to customize.

However, studying a decision on behalf of the participant is challenging, as extreme care must be taken to minimize any bias of that decision process. Here we discuss the factors and process of developing our study methodology in detail.

Motivation to Customize

In order for such a study to work, there must be a reasonable motivation for the user to customize. This means that the *enhanced behavior* that the customization enables must provide real benefits in terms of task performance or workload, in comparison to the *default behavior*. Unlike real-world scenarios, where the benefits of the enhanced behavior can propagate over long periods of time, lab experiments have a limited duration. As such the benefits need to be clear and immediate, so that it is to the user's advantage to customize. In particular, the enhanced behavior should not have a steep learning curve.

Between vs. Within Study Design

Because customization behavior involves a user's decision, any exposure to the *customization facility* or *enhanced behavior* could bias the user's decision to perform future customizations. If a user is exposed to multiple conditions, their decision to customize in later conditions could be influenced by experiences from the earlier conditions [9]. Thus, a between-subject design should be used.

Task Complexity

In real-world scenarios, customization is typically a secondary task that competes for the user's resources with the user's primary task. In the past, experiments used low cognitive demand tasks which illicit customization (*e.g.* [9]) or even instructed users to customize in order to compare the benefits of different personalization approaches (*e.g.* [2]). However, in our case, we wish to evaluate the *choice* to customize, so using a low cognitive demand task may artificially bias users towards customizing. To achieve a higher degree of external validity, the study may need to include tasks that vary in cognitive load. However, due to limited duration of lab experiments, it might not be possible to test different tasks. In such cases the study should include at least a single primary task with a reasonable cognitive load.

Customization Facility

Desktop applications allow for a wide variety of customization techniques [12]. In order to study customization behaviors, it is important that the customization facility used in the experimental task generalizes well. Although some in-place customization techniques have been explored [9, 19], customizations that result in significant benefits typically have an associated cost. For example, to define a new hotkey in Microsoft Word 2007, a user needs to access a dialog, 4 levels deep in the user interface. As such, for external validity, the *customization facility* should have some associated cost.

LESSONS FROM PILOT STUDIES

Due to all of the above-noted challenges, we conducted several pilot studies, to test various methodologies and study environments. Our initial methodology was based on Grossman *et al.*'s evaluation of hotkey usage [16]. A user was shown an image and could either select an associated menu item from the top of the screen, or use a customization dialog to set up a hotkey for that item (in the original work, the hotkey was already configured and available for use). However, we found that the mental demand of the enhanced behavior, using a hotkey, was too high, and as such participants consistently chose not to customize. It was apparent that this methodology did not provide enough *motivation to customize*.

To simplify the enhanced behavior, we introduced a local tool palette as the enhanced behavior. The users could access the palette faster than the default menu bar at the top of the screen. Because it offers a visual support for successful recall of customized operations, this approach required lower mental demand, but *all* participants were customizing, and reported that it was obvious that this would improve performance. Here, it became clear that the abstract image matching task we chose did not introduce enough *task complexity*.

As such, we introduced a more involved primary task to the methodology. After running several participants, the methodology seemed to provide a good balance, as advanced users chose to customize, while less advanced users did not. We settled on this methodology for our full experiment, which we now describe in detail.

EXPERIMENT

The goal of this experiment is to compare how exposure, awareness, and social factors influence customization behavior. We looked at how these customization factors influence the *breaking point*, the amount of customization that the user is willing to perform, and how these factors compare in terms of the benefit they provide to the user.

Participants

We recruited 64 participants (34 female, ages 18-63, median=27.5). The participants were in a wide variety of occupations. All were recruited through online classifieds ads and internal existing participants lists, and compensated with a $25 gift certificate.

Apparatus

We conducted the experiment on two HP Intel Xeon computers with 3.2GHz dual-processor and 12GB RAM, running Windows 7. Both machines were equipped with a 24-inch monitor set to 1920×1200 resolution, a standard optical mouse and QWERTY keyboard. Participants performed the task in separate rooms, and the investigators monitored their progress using a VNC-based remote desktop viewer. The experimental setup required two different rooms because for one of the conditions we required participants to come in with one of their friends and take part in the study at the same time. The experimental software was implemented using Java 5 SE.

Task

Figure 1 shows an overview of the experimental GUI and task procedure.

Main task. Each participant had to recreate a reference grid of items (Figure 1.III-e) by populating a working grid (Figure 1.III-d). Each item in the reference grid consisted of the following properties: shape (triangle, star, square, circle), fill color (yellow, green, red, blue), border (solid, dashed, dotted), and a foreground image (apple, banana, basket, clock, lemon).

Populating the grid. To recreate an item, the participant had to select a cell in the working grid, and then select a command corresponding to one of the four properties. The participants added item properties one by one to the active cell using one command for each property. Different items had different number of properties that the participant had to change, resulting in some properties being more frequent than others. A correct command caused the corresponding property to be added to the item in the grid. The grid selection was then cleared so that the participant had to select a cell to be edited before performing each command. This allowed tracking of trial times. A wrong command caused a 3 second penalty and the selection was cleared. No specific order was imposed for applying the properties to the cells, with the exception that the shape always had to be added first. The participant was not required to complete an item before selecting a different cell. Figure 1.II illustrates the completion of an item made of a star shape, a banana foreground image and blue fill color. Items appeared check-marked in both grids when completed.

Command selection. To perform the commands, the participants could select from one of the application menus (*default behavior*) corresponding to the property (Figure 1.IIIa) or the command palette (*enhanced behavior)* (Figure 1.IIIc). We expected a time saving of approximately 1 second every time the palette was used instead of the menu. All commands were available from the menus, but only a subset of commands was initially available from the palette. We intentionally pre-populated the palette with some of the commands, because leaving it empty may bias users towards customization [9]. Each property menu contained 12 commands (including distracters, which were never required to be used in the study). Figure 2 shows the default palette, split into four property categories. The default palette contained all the needed shapes, and a subset of the required other properties. Each palette section also included exactly one distracter that was never needed.

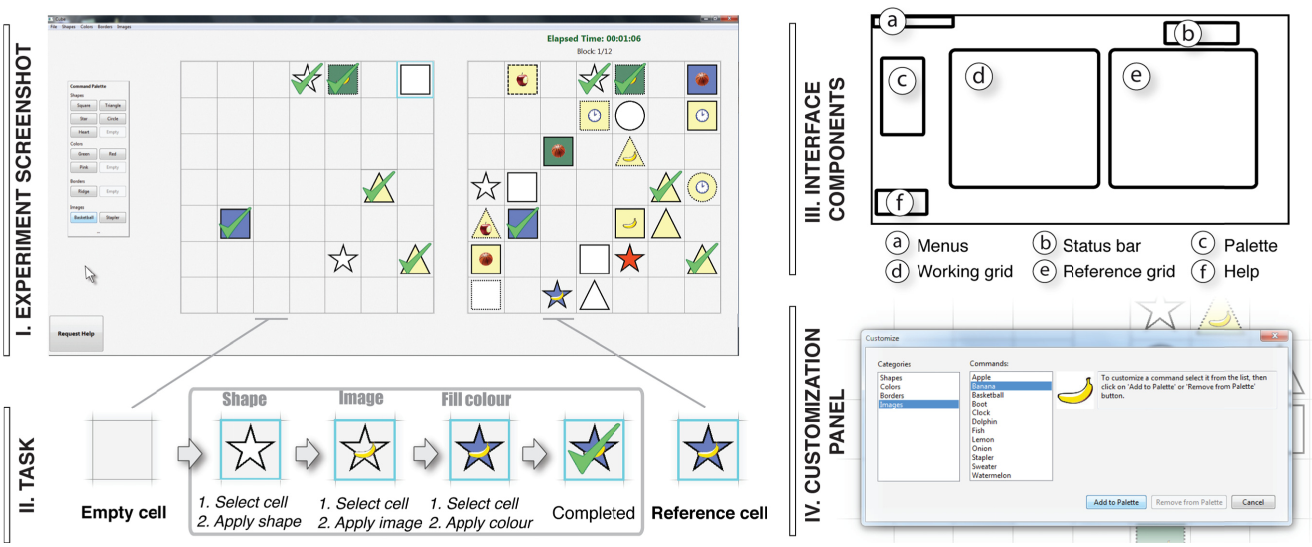

Figure 1. Experimental screenshot (I) with task (II), different components of the study interface (III), and customization panel (IV).

Customization facility. During the study the participants could customize the palette through a customization dialog (Figure 1.IV) invoked from the File menu to add or remove commands from the palette. For control purposes, the participants could customize only one command at a time. For a closer approximation to customization facilities in real software, we added a 10 second progress bar when loading the dialog.

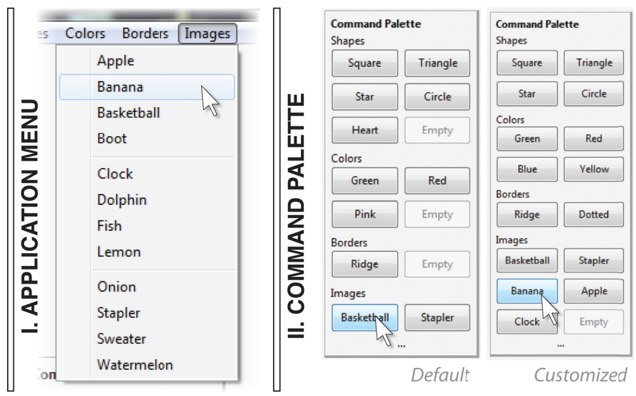

Figure 2. The application menu (I) and the default palette (II left) and a fully customized palette (II right) split into four property categories.

Procedure

Before the start of the experiment, an experimenter escorted the participants to a study room, but did not provide any verbal instructions about the task. Instead participants read the instructions once seated in the room. The instructions asked participants to perform the task as quickly and as accurately as possible. But the actual amount of time participants had to spend on the study was completely dependent on their own performance. As such, participants had an implicit motivation to complete the study efficiently.

Figure 3 shows an overview of the experimental procedure. Each participant completed 1 warm-up block and 12 study blocks. Each block consisted of a single reference grid. The set of items used to fill in the reference grid was the same across the study block, but the positions in the grid were randomly generated for each block. The participant had to execute exactly 10 correct commands in the warm-up block, and 60 correct commands in each of the study blocks. A block was completed only when all items in the grid were correctly recreated. During warm-up the participants had to use the application menu at least once, to become familiar with the default behavior.

Before the study, and after each block, the system informed users about the possibility to customize. The message was:

"All items can be accessed through the menu. The items available in the palette can be customized through the file menu."

The system provided no further instructions on how to customize. The participants were allowed to rest between blocks, with an enforced rest of at least 20 seconds.

Conditions. The experiment was a lab study with a between-subject design. The customization conditions were: *control*, *exposure*, *awareness*, and *social*. There were 16 participants in each condition. Each participant was randomly assigned to one of the four groups, but those participants who came in with a friend were only used in the *social* group. The conditions were as follows:

- *Control* – this condition provided only the mention of the existence of the customization feature (*i.e.* minimal awareness of the functionality).
- *Exposure* – this condition models high exposure to the customization feature. Participants in this group were forced to practice customizing one command during the warm-up, and then accessing that command from the palette. This was the only group that was ever instructed on how to customize. We included this condition because it provides the exposure through experiencing the customization facility.
- *Awareness* – this condition increases the awareness of the customization feature. Participants received a notification messages informing that customizing a specific command enables faster access (Figure 4). We chose this approach as it is similar to many past mixed-initiative systems.
- *Social* – this condition adds a social component to the raised awareness condition. In this condition, two friends completed the study at the same time in separate rooms. The participants received a similar message as for the *awareness* condition, but the message was crafted so that it said that the friend customized a command and was completing the task faster (Figure 4). The message was provided regardless of the friend's activity. This approach allowed us to test the impact of social influences on customization behavior.

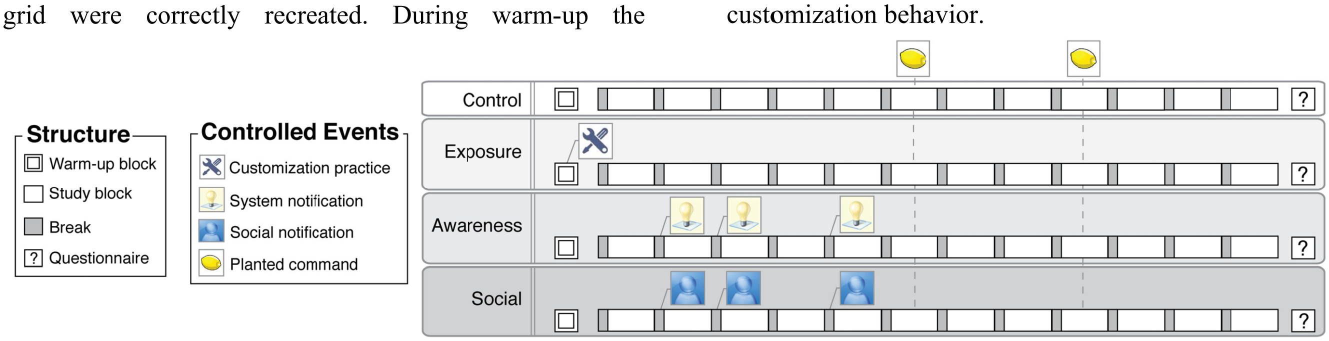

Figure 3. Overview of the experimental procedure for the four conditions.

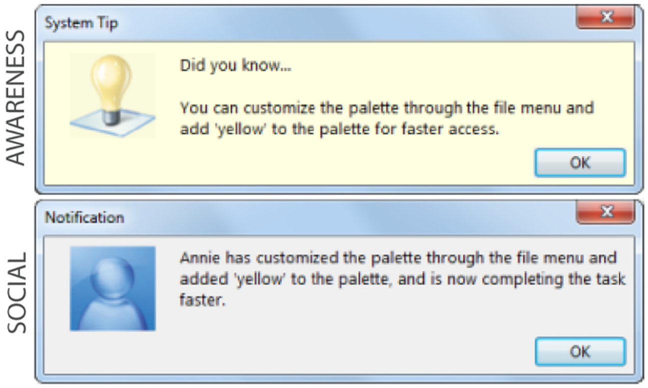

Figure 4. Awareness and social notification messages.

The participants in the *awareness* and *social* groups received notifications during the rest time after the first, second, and fourth grid. In the first message the system proposed the most frequent command that the participant had not customized at that point, in the second message it proposed the most frequent command that the participant had customized, and in the third message it proposed the next most frequent command that the user had not customized. If all commands were already customized, an arbitrary command was chosen.

Design

Each participant completed 12 grids × 60 commands each = 720 commands. The frequencies of commands were computed based on 30 random drawings from a Zipfian distribution of exponent 1 (relative frequency 1/rank) [39]. The frequencies were rounded to produce 7 frequencies (12, 6, 4, 3, 2, 2, 1). There were 14 commands that participants used in the experiment (2 assigned to each frequency). The command to frequency assignment was counterbalanced across participants, with each command mapped to a frequency once. The participant did not know the command frequencies upfront, but they became apparent as they were the same every block.

We also included one *planted* command, which only appeared twice across the entire study, to see how participants would respond to a command that would not be beneficial to customize. The *planted* command (lemon image) replaced one of the regular image commands in the 6^{th} and 9^{th} grids.

Additionally, the items in the grid had different properties and each item belonged to one of the four groups: items consisting of a shape only, a shape and 1 property, a shape and 2 properties, and a shape and 3 properties. The following default properties were assigned to a shape: white fill color, solid black border, and no foreground image. The participant never had to select any of the default properties as they were prepopulated on the item once a shape was added to the grid. The warm-up grid contained 4 items, one in each group, and a set of 24 items were used during the study, with 6 objects in each group.

Measures

We measured the performance time for individual commands, and the total study time, as defined below:

- *Command Time* - the average time between cell selection (click on a grid cell) to command execution (click on a command in a menu or in the palette). Only correct commands were considered.
- *Study Time* - the time it took to complete all trial grids (not including breaks).

We also measured when the participants first customized and the extent of customization which occurred:

- *Breaking Point* - the block in the experiment when the participant decided to customize the first command.
- *Customization Magnitude* - the number of customizations the participant performed in different blocks in the experiment.

Participant expertise levels. To provide additional insights into the data, we classified users as power or non-power users. To do so, participants completed a post-experiment questionnaire asking about their demographics, computer usage (*e.g.* amount of hours, type of tasks), and customization behavior. The customization behavior questions asked about the amount of customization (if any) and the type of customization the user would perform (e.g. personalizing Gmail account look, customizing Word command ribbon, modifying Word templates, creating Excel macros, etc.). This was done to account for different levels of customization experience [28]. Three of the authors performed data coding on participant's responses. The participants were grouped in two categories: power users (30 participants) and non-power users (34 participants). There was substantial agreement between the coders (Fleiss' κ=0.635). The split between power users and non-power users across conditions was relatively balanced (min=6, max=10, median=8).

RESULTS

In this section we present the results of our experiment. Unless otherwise stated, we analyzed parametric participant data (such as time to complete the study) with one-way and two-way ANOVAs. Type II ANOVA was used when data was mildly unbalanced. The pair-wise comparison was done using a Tukey's test. Non-parametric data, such as number of customizations we analyzed using a Kruskal-Wallis test and pair-wise comparison was done using a Mann-Whitney's U test. We used Fisher's exact test when comparing categorical data (*e.g.* number of participants that customized and those that did not).

Performance Time

One of the important design considerations in our methodology was to ensure that customization would benefit the user (*motivation to customize*). Accessing items from the command palette (mean=1.366s) was indeed faster than accessing an item from the menu (mean=3.226sec) ($F_{1,55}$=422.99, p<.001, η^2=0.13). Additionally, in all groups, participants who customized at least once completed the study faster than those who did not ($F_{1,56}$=23.47, p<.001, η^2=0.42), with average completion times of 38.59min and 49.03min respectively (Figure 5). It is also interesting to note that there was a difference in study completion times between conditions ($F_{3,56}$=4.26, p=.0087, η^2=0.23), where users in the *social* group completed the study faster than the *exposure* (p<.05) and *awareness* (p<.005) groups. Thus, the social factor might have influenced them to work faster.

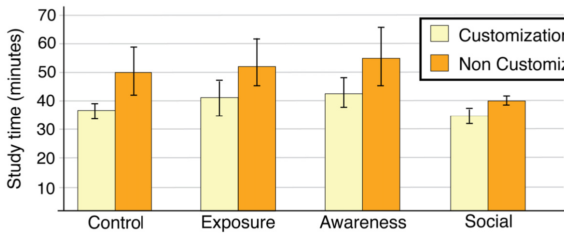

Figure 5. Study completion time per group and customization. The error bars represent 95% confidence interval.

Customization Strategies

Participants demonstrated different strategies, but at the high level, we looked at: (1) strategies of those that did not customize and (2) strategies of those that did customize. Across the 64 participants, 46 customized and 18 did not. Some users in the *control* group (N=3) chose to use only menus (1.1), but the majority of non-customizers used both the menu and the command palette (1.2). One participant who did not use the palette stated after the experiment that she prefers to "get into the flow" with one particular way of performing a task and dislikes switching.

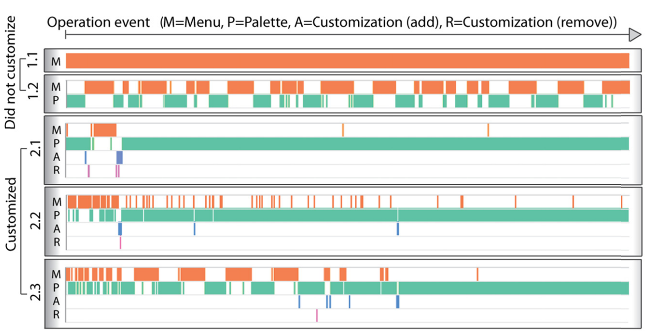

Figure 6. Examples of customization strategies of actual participants that did not customize (1.1 and 1.2) and those that customized study software (2.1, 2.2, and 2.3).

Strategies of those participants that customized can further be classified into: (2.1) early customization, (2.2) sparse customization and (2.3) later customization. The early customizers and late customizers tended to customize most of the commands in quick succession (usually within a single block), differing only in the time when they first started customization. The sparse customizers customized early in the experiment, but took time to customize all of the commands that they did, before switching to almost exclusively using the palette. Figure 6 shows examples of these customization strategies from actual participants.

Customization Behavior Between Conditions

We now look at the actual customization behaviors between groups. Figure 7 shows the number of participants that customized in the four conditions. In comparison to the *control*, the number of users who customized increased by 62.5% in the *exposure* and *awareness* conditions, and by 50% in the *social* condition. Despite the observable trend, the data only represents a single sample per user, and the difference did not reach statistical significance.

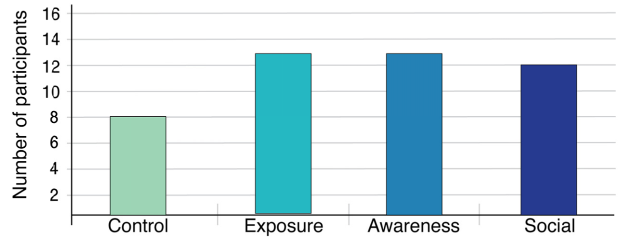

Figure 7. Number of participants out of 16 that customized study software by condition.

In the *control* group, 6 out of 8 power users and 2 out of 8 non-power users customized. This suggests that the study design, to a degree, resembles the real difference in customization behavior between power and non-power users [23]. We found a similar trend in the *awareness* group, and the difference in number of those who customized between non-power users (4 out of 7) and power users (9 out of 9) approached significance (p=.0625). We did not find significant difference between number of power and non-power users who customized in the other two groups. This may indicate that exposure and social factors are more effective for non-power users.

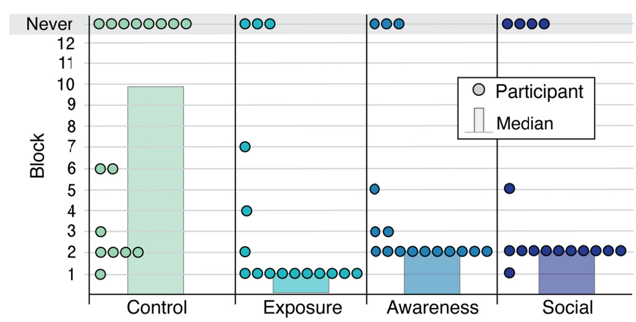

Figure 8. Median breaking point block per condition. Participants who did not customize are also included.

Breaking point. There was a significant difference between the breaking point between conditions ($\chi^2_{(3)}$=11.17, p=.0108) (Figure 8). Participants in the *control* group took longer to decide to customize than the *exposure* group (U=198, Z=2.76, p=.005, r=0.69). The *awareness* and *social* groups also appeared faster to customize than *control* and the difference approached significance (U=173, Z=1.81, p=.0652, r=0.45 and U=171.5, Z=1.764, p=.0696, r=0.44 respectively). The participants in the *exposure* group also first customized earlier than *awareness* (U=73.5, Z=2.14, p=.0329, r=0.54) and *social* (U=76, Z=2.06,

p=.0380, r=0.52) groups. This was likely because exposure occurred before the first block, whereas the awareness and social messages appeared before the second block. Overall, this provides empirical data that the evaluated factors can influence a user's *breaking point*.

Customization magnitude. Out of the 46 participants who customized at least one command, the percentage of those who customized all 7 of the commands was 57.45% (N=27). Some participants (N=17, 9 power users, 8 non-power users) removed some of the distractor commands from the palette. This is different from other studies that reported that participants rarely remove features [10]. This indicates that some users, not only switched to the *enhanced behavior*, but also tried to further improve their performance.

In order to further analyze the magnitude of customization, we grouped blocks in distinct periods in the experiment: start (blocks 1-4), middle (blocks 5-8), and end (blocks 9-12).

For all groups except control, most of the customizations occurred in the start period of the experiment (p<.05) (Figure 9). In the *control* group, the start and middle periods were not significantly different. This means that *exposure*, *awareness*, and *social* conditions can help users customize more items earlier, when customization has the most value.

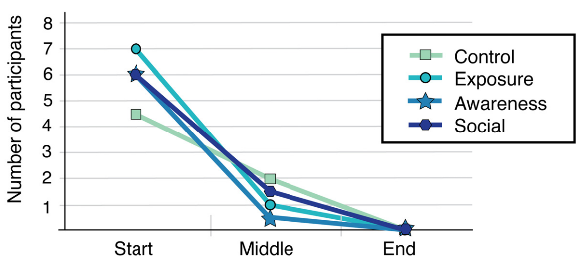

Figure 9. Median number of customizations by period and condition. The values do not include participants that have not customized.

Some participants also customized the *lemon*. The percentage of users that customized the *lemon* command in the middle period was: 0% in the *control* group, 25% in the *exposure* group, 0% in the *awareness* group, and 6.25% in the *social* group. The differences approached significance (p=.0652). One other participant from the *awareness* group (6.25%) customized *lemon* in the end period. This could mean that some participants (mostly in the *exposure* group) acted impulsively and simply repeated the learned action to customize without considering the value of customization.

Subjective Results

After the experiment we asked the participants about how much impact the assigned condition had on their strategy (*e.g.* "During the study you have received number of notifications about what your friend was doing. To which extent did this impact your strategy?"). Responses were measured on a scale from 1 to 7, where 1 was very low impact and 7 was very high impact. The mean results were 4 in *exposure* and *awareness*, and 5 in *social*. The differences were not significantly different (p=.452). However, the open-ended questions highlighted some interesting differences between conditions.

The participants in the *exposure* group saw the training as a means to raise their awareness of the customization facility. However, none of them commented that exposure helped them in realizing that customization was easy to perform:

"If it had not been for the information I derived during the warmup period, I wouldn't have customized my palette to the elements that appeared most often." –P12$_{exposure}$

"Once I found out I can customize, I thought about when it is best to customize and when it was not." –P16$_{exposure}$

For some users, exposure did not trigger customization because they were unsure of the benefits of the enhanced behavior:

"At first I didn't bother [to customize], as I didn't see it as a factor, thinking that there would be so many grid boxes to replicate that choosing customizations would just take up space and cost time." –P8$_{exposure}$

For participants in the *awareness* group, raising the awareness of customization facility was enough to trigger customization:

"At first I was unaware of the ability to customize the palette. After I received that advice from the system, I realized the advantage of having all the most commonly used commands on the palette and quickly added them." –P14$_{awareness}$

However, the content of the message was less useful to some power users. They were able to understand the benefit of the *enhanced behavior* even before the notification:

"The tips were never particularly important, only the knowledge I could customize, which I did at a later while with my own decisions." –P11$_{awareness}$ (power user)

However, non-power users found this information helpful:

"It was as if the system was giving clues as to what was going to be needed to do the next task." –P5$_{awareness}$ (non-power user)

Most participants felt that the social message increased their awareness of the customization facility. It also made them perform the task faster in order to compete with their friend.

"I didn't even think about seeing if I could customize the palette until I noticed that my friend was doing it. It sent me in the right direction." –P15$_{social}$

"I get very easily competitive, so seeing that he was performing tasks faster made me want to do better myself. Seeing that the customizations helped his strategy made me take the time to make my own customizations to enable me to work to my full capacity to complete the task." –P16$_{social}$

The social message could have also been a tipping point for some, as it increased their confidence that they would be able to learn how to use the customization facility.

"At first I did not want to change the option. I figured it would take too much work. But when I saw that my friend changed it, I thought I might as well try it once." –P10 $_{social}$

"During the first round I was not adventurous enough to spend time discovering how to use/customize the pallet. After seeing my friend had, I decided to attempt to customize my own, and discovered it was a fairly simple process." – P1$_{social}$

However, for some participants in both *awareness* and *social* groups, the notifications had little impact. In fact some chose to disregard the messages thinking the content had no importance, while for others the messages added negative pressure, and may have inhibited customization:

"I didn't want to get stressed out by what he was doing. I really didn't care what he was doing to be honest!" –P5$_{social}$

"I didn't want to waste time [finding out how to customize]. I tend to be rather competitive, so it just added extra pressure to perform. [My friend] is competitive too, so I like to beat him if I can." –P7$_{social}$

DISCUSSION

Through our analysis, we have identified that the factors we tested have a significant influence on users' customization behavior. In this study, this was mainly manifested in changing when a user chose to customize. While not significant, it also seemed to influence whether or not a user chose to customize. These are encouraging results.

The analysis revealed only subtle differences in the impact of the three test conditions. Larger scale studies, with more participants, may be required to more accurately measure the relative impact of these factors. Most prominent was that in the *social* condition, users performed the task significantly faster than in other conditions. This could be due to competition rather than benefits of customization. Even so, in addition to changing the customization behavior this approach could be used to increase user performance.

We also found that the factors might have a different effect on power and non-power users. In the *control* group, only 25% non-power users customized, and in the *awareness* group, only 57%. On the other hand, in the *exposure* and *social* conditions, more non-power users customized (80% and 78% respectively). This may indicate that awareness is a weaker influence for non-power users.

In an actual application, exposure to the customization facility could be achieved through occasional enforced training. But it could also be combined with social influences, where the system notifies the user about her friend's or colleague's behavior and offers to guide her through the customization process.

It was interesting to note that some participants chose not to customize even in the test conditions. While we expected this behavior in the *control* group, we found various reasons for such behavior in other groups. In the *exposure* group this could be due to participants being overly reluctant to changing their behaviors [4]. Providing a rationale for customization might help [3], but it remains unclear how behavior of such users could be changed, and more research is required. In the *awareness* and *social* groups some participants did not customize simply because they chose to disregard the messages. As such, other techniques to expose users to awareness and social factors, besides system messages, should be explored. An ambient display of such information, which a user would not be as rushed to dismiss, could be one potential solution [25].

The *social* condition might have an adverse effect on some users, as the pressure to perform faster could inhibit their willingness to explore customization options. Future research should explore how social factors affect users with different personalities. However, as users were under time pressure in the lab study, it remains an open question if such adverse effects would occur in a more relaxed setting.

A secondary contribution of our work is that we show how sensitive customization studies are to the particulars of the study methodology. Small changes made after each of our pilots had large effects on user behaviors. Our experiences, and our methodology considerations discussion, provide important lessons for future customization studies.

In summary, each of the factors we tested impacted customization behaviors, and some trade-offs between the techniques appeared. In order to design successful customization facilities, designers should consider the strengths and weaknesses of these factors. However, some of these influences could potentially work together (*e.g.* social influence on top of a mixed-initiative system) to remedy some of the weaknesses and enhance the strengths.

CONCLUSION AND FUTURE WORK

In this paper we demonstrated a methodology for personalization studies which require a user choice. We then used this methodology to design and execute a user study. The results of the study exposed different customization strategies and differences in how users respond to increased exposure and awareness, and social influences. We showed that these three factors can decrease the time to the *breaking point*, and that software can in fact control such factors in order to change user behavior.

Based on these findings we proposed directions in which future research can improve customization facilities. While our results may generalize beyond the specific facility and enhanced behavior from our methodology, the factors should be evaluated against different customization approaches [12]. Also, our findings provide grounding for follow-up field studies with more complex and real-world software applications to test whether our results would extend to such systems. As we discussed, lab studies have inherent challenges when running studies that involve user choice. It would be interesting to develop field study methodologies to test how closely our results would be replicated in actual usage scenarios. Nevertheless, our findings provide insights into customization factors and allow for more focused field studies of customization factors that can be controlled by software.

REFERENCES

1. Brusilovsky, P. and Schwarz, E. User as student: towards an adaptive interface for advanced web-based applications. *User Modeling*, (1997), 177-188.
2. Bunt, A., Conati, C., and McGrenere, J. Supporting interface customization using a mixed-initiative approach. In *Proc. IUI '07*. ACM (2007), 92-101.
3. Bunt, A., McGrenere, J., and Conati, C. Understanding the Utility of Rationale in a Mixed-Initiative System for GUI Customization. In *Proc. UM '07*. (2007), 147-156.
4. Carrol, J. M., and Rosson, M. B. Paradox of the active user. In *Interface thought: Cognitive aspects of human-computer interaction*. Cambridge (1987).
5. Clark, B., Matthews, J. Deciding Layers: Adaptive Composition of Layers in a Multi-Layer User Interface. In *HCI International*, Volume 7, July 2005.
6. Cockburn, A., Gutwin, C., and Greenberg, S. A predictive model of menu performance. In *Proc. CHI '07*. ACM (2007), 627-636.
7. Debevc, M., Meyer, B., Donlagic, D., and Svecko, R. Design and evaluation of an adaptive icon toolbar. User Mod. and User-Adap. Inter., 6, 1, (1995), 1-21.
8. Dourish, P., Lamping, J., and Rodden, T. Building bridges: customisation and mutual intelligibility in shared category management. In *Proc. GROUP '99*. ACM (1999), 11-20.
9. Findlater, L. and McGrenere, J. A comparison of static, adaptive, and adaptable menus. In *Proc. CHI '04*. ACM (2004), 89-96.
10. Findlater, L. and McGrenere, J. Evaluating reduced-functionality interfaces according to feature findability and awareness. In *Proc. INTERACT'07*. Springer-Verlag (2007), 592-605.
11. Findlater, L. and McGrenere, J. Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces. In *Proc. CHI '08*. ACM (2008), 1247-1256.
12. Findlater, L. and McGrenere, J. Beyond performance: Feature awareness in personalized interfaces. Int. J. Hum.-Comput. Stud. 68, 3 (March 2010), 121-137.
13. Findlater, L., Moffatt, K., McGrenere, J., and Dawson, J. Ephemeral adaptation: the use of gradual onset to improve menu selection performance. In *Proc. CHI '09*. ACM (2009), 1655-1664.
14. Gajos, K. Z., Czerwinski, M., Tan, D. S., and Weld, D. S. Exploring the design space for adaptive graphical user interfaces. In *Proc. AVI '06*. ACM (2006), 201-208.
15. Greenberg, S. and Witten, I. Adaptive personalized interfaces: A question of viability. Behav. Inf. Tech. 4, 1, (1985), 31-45.
16. Grossman, T., Dragicevic, P., and Balakrishnan, R. Strategies for accelerating on-line learning of hotkeys. In *Proc. CHI '07*. ACM (2007), 1591-1600.
17. Grossman, T., Fitzmaurice, G., and Attar, R. A survey of software learnability: metrics, methodologies and guidelines. In *Proc. CHI '09*. ACM (2009), 649-658.
18. Krisler, B. and Alterman, R. Training towards mastery: overcoming the active user paradox. In *Proc. NordiCHI '08*. ACM (2008), 239-248.
19. Kurlander, D. and Feiner, S. A history-based macro by example system. In *Proc. UIST '92*. ACM (1992), 99-106.
20. Mackay, W.E. Patterns of sharing customizable software. In *Proc. CSCW '90*. ACM (1990), 209-221.
21. Mackay, W.E. Triggers and barriers to customizing software. In *Proc. CHI '91*. ACM (1991), 153-160.
22. MacLean, A., Carter, K., Lövstrand, L., and Moran, T. User-tailorable systems: pressing the issues with buttons. In *Proc. CHI '90*. ACM (1990), 175-182.
23. Marathe, S.S. Investigating the psychology of task-based and presentation-based UI customization. In *Proc. CHI EA '09*. ACM (2009), 3129-3132.
24. Marathe, S. and Sundar, S.S. What drives customization?: control or identity?. In *Proc. CHI '11*. ACM (2011), 781-790.
25. Matejka, J., Li, W., Grossman, T., and Fitzmaurice, G. CommunityCommands: command recommendations for software applications. In *Proc. UIST '09*. ACM (2009), 193-202.
26. McGrenere, J., Baecker, R.M., and Booth, K.S. An evaluation of a multiple interface design solution for bloated software. In *Proc. CHI '02*. (2002), 164-170.
27. Mitchell, J. and Shneiderman, B. Dynamic versus static menus: an exploratory comparison. SIGCHI Bull. 20, 4 (April 1989), 33-37.
28. Mørch, A. Three levels of end-user tailoring: customization, integration, and extension. In *Computers and design in context*. MIT Press (1997), 51-76.
29. Page, S.R., Johnsgard, T.J., Albert, U., and Allen C.D. User customization of a word processor. In *Proc. CHI '96*. ACM (1996), 340-346.
30. Peres, S.C., Tamborello II, F.P., Fleetwood , M. D., Chung, P., and Paige-smith, D. L. Keyboard shortcut usage: The roles of social factors and computer experience. *Hum. Fact. & Ergo. Soc., 48.* (2004), 803-807.
31. Scarr, J., Cockburn, A., Gutwin, C. and Quinn, P. Dips and ceilings: understanding and supporting transitions to expertise in user interfaces. In *Proc. CHI '11*. ACM (2011), 2741-2750.
32. Sears, A. and Shneiderman, B. Split menus: effectively using selection frequency to organize menus. ACM Trans. Comput.-Hum. Interact. 1, 1 (1994), 27-51.
33. Shneiderman, B. and Plaisant, C. Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Ed.). Pearson Addison Wesley (2004).
34. Stuerzlinger, W., Chapuis, O., Phillips, D., and Roussel, N. User interface façades: towards fully adaptable user interfaces. In *Proc. UIST '06*. ACM (2006), 309-318.
35. Tsandilas, T. and schraefel, m.c. An empirical assessment of adaptation techniques. In *CHI EA '05*. ACM (2005), 2009-2012.
36. Thomas, C.G. and Krogsæter, M. An adaptive environment for the user interface of Excel. In *Proc. IUI '93*, ACM (1993), 123-130.
37. Twidale, M.B. Over the Shoulder Learning: Supporting Brief Informal Learning. *Comput. Supported Coop. Work 14, 6* (December 2005), 505-547.
38. Venkatesh, V., Morris, M.G., Davis, F.D., and Davis, G.B. User Acceptance of Information Technology: Toward a Unified View. *MISQ (27:3)*, 2003, 425-478.
39. Zipf, G. *Human behavior and the principle of least-effort*. Addison-Wesley (1949).