Improving the Performance of Motor-Impaired Users with Automatically-Generated, Ability-Based Interfaces

Krzysztof Z. Gajos
Computer Science & Eng.
University of Washington
Seattle, WA 98195 USA
kgajos@cs.washington.edu

Jacob O. Wobbrock
The Information School
University of Washington
Seattle, WA 98195 USA
wobbrock@u.washington.edu

Daniel S. Weld
Computer Science & Eng.
University of Washington
Seattle, WA 98195 USA
weld@cs.washington.edu

ABSTRACT

We evaluate two systems for automatically generating personalized interfaces adapted to the individual motor capabilities of users with motor impairments. The first system, SUPPLE, adapts to users' capabilities indirectly by first using the ARNAULD preference elicitation engine to model a user's preferences regarding how he or she likes the interfaces to be created. The second system, SUPPLE++, models a user's motor abilities directly from a set of one-time motor performance tests. In a study comparing these approaches to baseline interfaces, participants with motor impairments were 26.4% faster using ability-based user interfaces generated by SUPPLE++. They also made 73% fewer errors, strongly preferred those interfaces to the manufacturers' defaults, and found them more efficient, easier to use, and much less physically tiring. These findings indicate that rather than requiring some users with motor impairments to adapt themselves to software using separate assistive technologies, software can now adapt itself to the capabilities of its users.

Author Keywords

SUPPLE, SUPPLE++, ARNAULD, motor impairments, ability-based user interfaces

ACM Classification Keywords

H.5.2 Information Systems Applications: Information Interfaces and Presentation—*User Interfaces*; K.4.2 Computers and Society: Social Issues—*assistive technologies for persons with disabilities*

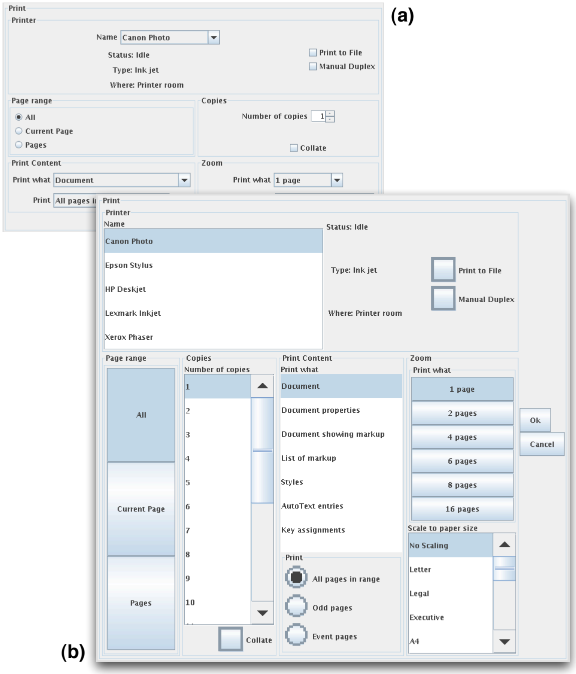


Figure 1. (a) The default interface for a print dialog. (b) A user interface for the print dialog automatically generated for a user with impaired dexterity based on a model of her actual motor capabilities.

INTRODUCTION

Computer use is a continually increasing part of our lives at work, in education, and when accessing entertainment and information. Thus the ability to use computers efficiently is essential for equitable participation in an information society. But users with motor impairments often find it difficult or impossible to use today's common software applications [2]. While some may contend that the needs of these users are adequately addressed by specialized assistive technologies, these technologies, while often helpful, have two major shortcomings. First, they are often abandoned, because of their cost, complexity, limited availability and need for ongoing maintenance [5, 6, 16, 21]. In fact, it is estimated that only about 60% of the users who need assistive technologies actually use them [6]. Second, assistive technologies are designed on the assumption that the user interface, which was designed for the "average user," is immutable, and thus users with motor impairments must adapt *themselves* to these interfaces by using specialized devices [15, 26].

A preferable solution would be to adapt user interfaces to the actual abilities of individual users with motor impairments. Unfortunately, because of the great variety of individual capabilities among such users [2, 13, 15, 17], manually designing interfaces for each one of them is impractical and not scalable. In this paper, we show that user interfaces can *automatically adapt themselves* to users' capabilities, allowing users access to custom interfaces fine-tuned to their abilities.

This paper evaluates two systems that automatically generate user interfaces customized to a user's individual capabilities. The first system, SUPPLE [8], adapts to user's capabilities indirectly by first using the ARNAULD preference elicitation engine [9] to model a user's preferences regarding how he or she likes the interfaces to be created. The second, SUPPLE++ [10], relies on its built-in Ability Modeler to model a user's motor abilities directly through a set of one-time motor performance tests.

The results of our study, which compared these two approaches to baseline interfaces with 11 motor-impaired and 6 able-bodied participants, show that participants with motor impairments were significantly faster, made many fewer errors, and strongly preferred the automatically-generated personalized interfaces, particularly those generated by SUPPLE++, over the baselines. Our results demonstrate that users with motor impairments can perform well with conventional input devices (e.g., mice or trackballs) if provided with interfaces that accommodate their unique motor capabilities. We also show that automatic generation of user interfaces based on users' motor abilities is feasible and that the resulting interfaces are an attractive alternative to manufacturers' defaults.

In the remainder of the paper, we present a brief overview of the systems tested. Next, we describe the first part of the study, where we used SUPPLE++'s Ability Modeler to build models of participants' motor capabilities and ARNAULD to elicit their preferences regarding user interface design. We then present the main experiment, where we compare the performance and satisfaction of participants using automatically generated user interfaces to the manufacturers' default interfaces. We follow with a discussion of the results and conclusions.

RELATED WORK

A number of specialized software applications have been developed with a particular subset of the motor-impaired population in mind. For example, EyeDraw [12] provides a convenient way for eye-tracker users to create art, while VoiceDraw [11] allows people to paint strokes with non-verbal vocalizations.

While a number of systems address the scalability challenge of adapting any application to the needs of users with vision impairments (e.g., [3]), few do it for users with motor impairments. For example, Mankoff et al. [19] created a system that automatically modifies web pages for the needs of users with severely restricted motor capabilities. Meanwhile, Input Adapter Tool [4] offers the possibility of modifying user interfaces of any application written in Java Swing to improve the accessibility for users with motor impairments. However, this system can generally only replace widgets with similarly-sized alternatives (e.g., text boxes with combo boxes) and cannot affect the organization of the interface or the sizes of the interactors.

Others, arguing that user interfaces need to be assembled dynamically for individual users [24], have developed components of necessary infrastructure [22, 23], but no general artifact seems to be available for evaluation at this time. Automatic generation of user interfaces offers the promise of providing personalized interfaces on-the-fly, but most systems, such as [20], need to be manually modified for each new platform or interaction style.

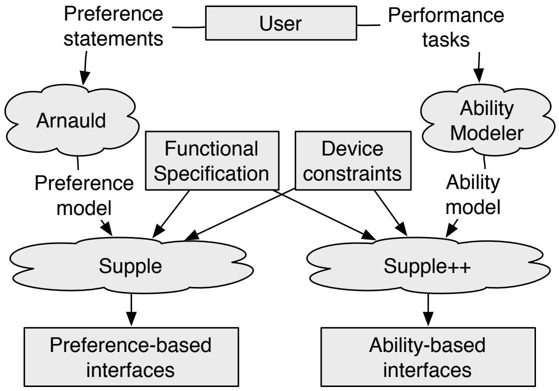


Figure 2. Relationship between the systems studied in this paper.

BACKGROUND AND OVERVIEW

In this section, we briefly review the three previous projects, SUPPLE, ARNAULD, and SUPPLE++, upon which this work relies; we then outline our experiment.

SUPPLE [8] automatically generates user interfaces, taking as inputs device-specific constraints, such as screen size and a list of available interactors, a typical *usage trace*, a *functional specification* of the interface, which describes the types of information that need to be communicated between the application and the user, and a *cost function*. SUPPLE performs decision-theoretic optimization, using the cost function to guide search for the interface with lowest estimated cost, which satisfies all the device constraints. One can make SUPPLE accommodate various objectives by modifying the cost function. Specifically, the ARNAULD system [9] can elicit and capture a user's *preferences* allowing SUPPLE to generate interfaces (even for previously unseen applications) that are likely to capture the user's general GUI design preferences. In contrast, SUPPLE++'s Activity Modeler [10] builds an explicit model of the user's actual *motor capabilities* by asking the user to complete a one-time motor performance test. By using this *ability model* as a cost function, SUPPLE++ generates the interface, which is predicted to let the user accomplish a set of typical tasks in the least amount of time. Figure 2 illustrates the relationship between these systems.

The major technical shortcoming of the first SUPPLE++ prototype [10] was its inability to accurately model list selection times. At first, SUPPLE++ modeled list selections in terms of its individual sub-operations (pointing, dragging, clicking). Although SUPPLE++ accounted for the multiple ways to operate a typical list widget, it modeled scroll bars in the same way as other dragging tasks. Unfortunately, dragging a scroll bar can take much longer than other dragging tasks, because the target of the drag operation isn't visible at the start of movement. Gajos et al. [10] therefore extended SUPPLE++ to include explicit list selection tasks during the ability elicitation phase, thus learning a more detailed model, which accounts for visual verification during scrolling. This

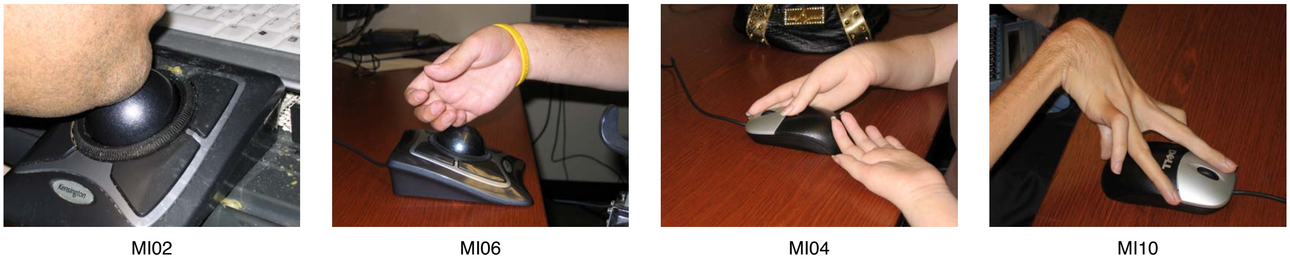


Figure 3. Different strategies employed by our participants to control their pointing devices (MI02 uses his chin).

extension was never formally evaluated, however, so in this paper we verify the benefits of the direct modeling of list selection times before using it to generate user interfaces for the study.

In the next section, we report on the first phase of the study where we used Arnauld to build models of our participants' preferences and Supple++'s Activity Modeler to construct models of their actual motor abilities. In the following section, we describe the main experiment, in which we compare participants' performance while using automatically generated user interfaces for three different applications and the corresponding manufacturers' defaults.

Because some participants found the Supple++ ability test somewhat tiring, and to allow for proper preparation time, we conducted the two parts of the study on separate days for each participant.

ELICITING PERSONAL MODELS

In this section, we describe the first part of the study, where we used Arnauld to build a model of participants' preferences and Supple++ to model their motor abilities.

Method

Participants

Altogether, 11 participants with motor impairments (age: 19-56, mean=35; 5 female) and 6 able-bodied participants (age: 21-29, mean=24; 3 female) recruited from the Puget Sound area took part in the study. The abilities of participants with motor impairments spanned a broad range (Table 1) and they used a variety of approaches to control the pointing device (Figure 3). All but one reported using a computer multiple hours a day and all reported relying on the computer for some critical aspect of their lives (Table 2).

Apparatus

We used an Apple MacBook Pro (2.33GHz, 3Gb RAM) for all parts of the study. Most participants were tested at our lab using an external Dell UltraSharp 24" display running at 1920×1200 resolution, but 3 of the 11 motor-impaired participants chose to conduct the experiment at an alternative location of their choosing; in these cases, we used the built-in 15" display running at the 1440×900 resolution.

Each participant had the option of adjusting the parameters of their chosen input device (e.g., tracking speed, button functions). Additionally, we offered the participants with motor impairments the option to use any input device of their choosing, but all of them chose to use either a Dell optical mouse or a Kensington Expert Mouse trackball (Table 1). All able-bodied participants used a mouse. The same equipment with the same settings was used in both parts of the experiment by each participant.

Participant	*Health Condition*	*Device Used*	*Controlled with*
MI01	Spinal degeneration	Mouse	hand
MI02	Cerebral Palsy (CP)	Trackball	chin
MI03	Friedrich's Ataxia	Mouse	hand
MI04	Muscular Dystrophy	Mouse	two hands
MI05	Parkinson's	Mouse	hand
MI06	Spinal Cord Injury	Trackball	backs of the fingers
MI07	Spinal Cord Injury	Trackball	bottom of the wrist
MI08	Undiagnosed; similar to CP	Mouse	fingers
MI09	Spinal Cord Injury	Trackball	bottom of the fist
MI10	Dysgraphia	Mouse	hand
MI11	Spinal Cord Injury	Mouse	hand

Table 1. Detailed information about participants with motor impairments (due to the rarity of some of the conditions, in order to preserve participant anonymity, we report participant genders and ages only in aggregate).

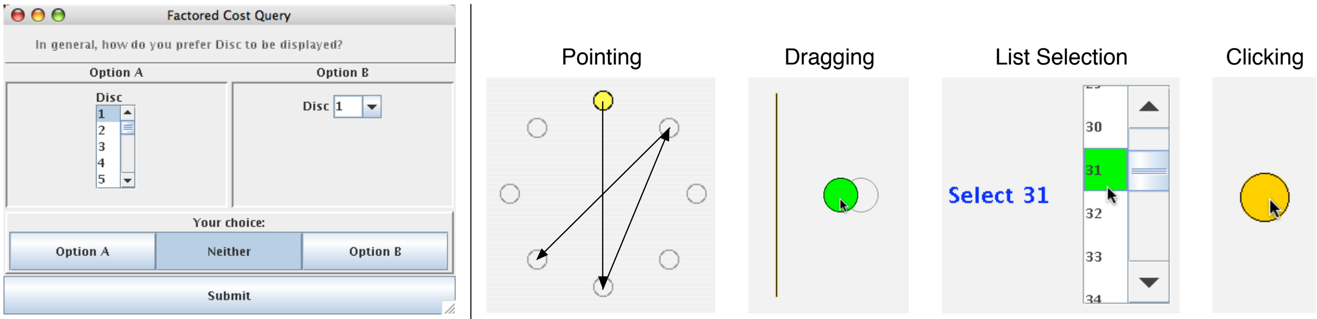
Do you rely on being able to use a computer for...	*# out of 11*
Staying in touch with friends, family or members of your community?	10
School or independent learning?	7
Work?	6
Entertainment?	11
Shopping, banking, paying bills or accessing government services?	10

Table 2. Numbers of participants with motor impairments depending on a computer for different activities.

Preference Elicitation Tasks

We used Arnauld to elicit participants' preferences regarding presentation of graphical user interfaces. Arnauld supports two main types of interactions: *active elicitation* and *example critiquing*.

Active elicitation is a computer-guided process, where participants are presented with queries showing pairs of user interface fragments and asked which, if either, they prefer. The two interface fragments are functionally equivalent but differ in presentation. The fragments are often as small as a single element (see Figure 4a) but can be a small subset of an application or an entire application. The queries are generated automatically based on earlier responses from the participant, so each participant saw a different set of queries. The interface fragments used in this study came from two applications: a classroom controller application (which included controls for three dimmable lights, overhead projector with selectable inputs, a motorized screen and a ventilation system; see [8]-Figure 2) and a stereo controller applica-



Factored Cost Query

In general, how do you prefer Disc to be displayed?

(a) An example of a query used during the active elicitation part of the preference elicitation.

(b) Four task types used to measure participants' motor capabilities

Figure 4. Interactions for building personalized models of the participants

tion (with master volume control, CD player, tape deck and a tuner; see [7]-Figure 2). These applications were unrelated to those used in the next phase of this experiment.

During the example critiquing phase, the participants were shown what interfaces SUPPLE would generate for them for the classroom and stereo applications. The participants were then offered a chance to suggest improvements to those interfaces. The experimenter would use SUPPLE's customization capabilities [7] to change the appearance of those interfaces. These customization actions were used as additional input by ARNAULD. If a participant could not offer any suggestions, the experimenter would propose modifications. The original and modified interfaces would then be shown to the participant. Participant's acceptance or rejection of the modification would be used as further input to ARNAULD.

Ability Elicitation Tasks

We used the SUPPLE++ Ability Modeler (extended to use the list selection tasks) to build a model of each participant's motor capabilities. We modified the task set used in [10] to accommodate the additional list selection tasks.

The four types of tasks used to elicit each participant's motor abilities are all illustrated in Figure 4b. They were:

- ***Pointing.*** We used a set of pointing tasks based on the ISO 9241-9 standard [14] where we varied target size (10-90 pixels at 6 discrete levels), distance (25-675 pixels, 7 levels), and movement angle (16 distinct, uniformly spaced angles).
- ***Dragging.*** We used a set of reciprocal dragging tasks where the dragged object's movement was constrained to be in one dimension (horizontal or vertical), emulating the behavior of standard GUI components such as scroll bar elevators or sliders. We varied target size (10-40 pixels, 3 levels), distance (100 or 300 pixels) and direction (up, down, left, right).
- ***List Selection.*** We asked our participants to alternately select two numbers in the list of consecutive numbers. Both numbers to be selected were placed sufficiently far from the end of the range so that they could not be accessed when the scroll bar was moved all the way to the top or to the bottom. We varied the height of the scroll window (5, 10, or 15 items), the distance in number of items between items to be selected (10-120, 7 levels), and the minimum size of any clickable element, such as list cells, scroll buttons, scroll bar elevator, or scroll bar width (15, 30, or 60 pixels).
- ***Multiple Clicking.*** We used 5 targets of diameters varying from 10 to 60 pixels to measure the rate at which participants could perform multiple clicks within targets of various sizes.

Procedure

At the beginning of the session, participants had a chance to adjust input device settings (e.g., tracking speed) and the physical setup (e.g., chair height, monitor position). We then proceeded with preference elicitation followed by ability elicitation, encouraging the participants to rest whenever necessary. At the end of the session, we administered a short questionnaire asking participants to asses how mentally and physically demanding the two elicitation methods were (on a 7-point Likert scale), and to state their overall preference.

Preference elicitation took 20-30 minutes per participant. Ability elicitation took about 25 minutes for able-bodied participants and between 30 and 90 minutes for motor-impaired participants. We analyzed subjective Likert scale responses for the main effect of elicitation method using ordinal logistic regression [29].

Results

Subjective Ratings

On a Likert scale (1-7) for how *mentally* demanding (7 = very demanding) the two tasks were, participants ranked ability elicitation as a little more mentally demanding (2.82) than preference elicitation (2.24), but the difference was not significant ($\chi^2_{(1,N=34)}$=1.62, n.s). They did see ability elicitation as much more *physically* demanding (4.73) than the other method (1.82) and this difference was significant ($\chi^2_{(1,N=34)}$=51.23, $p < .0001$).

When asked which of the two personalization approaches they would prefer if they had to choose one (assuming equivalent results), 9 of 11 motor-impaired participants preferred the preference elicitation (6 strongly). The two motor impaired participants who somewhat preferred ability-elicitation commented that it felt like a game.

Among able-bodied participants, 3 strongly preferred preference elicitation, 2 somewhat preferred ability-elicitation, and 1 had no preference for either approach.

Preference Model

Between 30 and 50 active elicitation queries and 5 to 15 example critiquing answers were collected from each participant. Between 51 and 89 preference constraints (mean=64.7) were recorded for each participant (some example critiquing responses could result in several constraints being recorded for a single participant response [9]). On average, the cost functions generated by ARNAULD were consistent with 92.5% of the constraints generated from any one participants' responses. This measure corresponds to a combination of two factors: consistency of participants' responses and the ability of SUPPLE's cost function to capture the nuances of participant's preferences. While this result cannot be used to make conclusions about either the participants or the system alone, it does give us confidence that the resulting interfaces will reflect users' stated preferences fairly accurately.

List Selection Model

We analyzed the fit of both the old and the new list selection modeling approaches to the data collected from the list selection tasks. For the component-based approach used originally in SUPPLE++, the mean R^2 for both groups of participants was only .09 (ranging from .00 to .36). In contrast, the direct model built from the data collected from the list selection tasks in this study had a mean R^2 fit of .61 (range: .39-.84) for motor-impaired and .67 (.49-.76) for able-bodied participants. In light of these results, we decided to use the new direct model for generating ability-based interfaces in the second part of the experiment.

EXPERIMENT

In this section, we describe an experiment that evaluated the effects on performance and satisfaction of automatically generated personalized interfaces compared to the baseline versions of those interfaces. Both types of personalized interfaces were tested: those based on participants' stated preferences and those based on their measured abilities.

Method

Participants and Apparatus

The same participants took part in both phases of the study, using the same equipment configurations.

Tasks

We used three different applications for this part of the study: a font formatting dialog box from Microsoft Word 2003, a print dialog box from Microsoft Word 2003, and a synthetic application (see Figure 8, also used by [10]). The first two applications were chosen because they are frequently used components from popular productivity software. The synthetic application was used because it exhibits a variety of data types typically found in dialog boxes, some of which were not represented in the two other applications (for example, approximate number selections, which can be represented in an interface with a slider or with discrete selection widgets).

For each application, participants used three distinct interface variants: *baseline*, *preference-based*, and *ability-based*. The baseline interfaces for the font formatting and print dialog boxes were the manufacturer's defaults re-implemented in SUPPLE to allow for instrumentation, but made to look very similar to the original (see Figure 5). For the synthetic application, we used the baseline from [10] (see Figure 8 left) because it has a very "typical" design for a dialog box: it is compact, and relatively uncluttered.

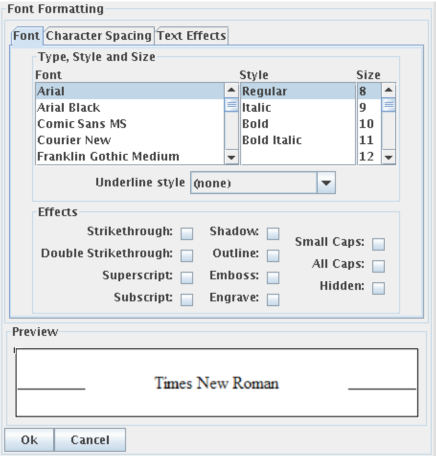


Figure 5. The baseline variant for the font formatting application. It was designed to resemble the implementation in MS Office XP. Two color selection widgets were removed and the preview pane was not functional.

Both the preference- and the ability-based interface variants were automatically generated individually for each participant using individual preference and ability models elicited during the first meeting with the participant.

For the automatically generated user interfaces, we set a space constraint of 750×800 pixels for print and synthetic applications and 850×830 pixels for the font formatting application (see Figures 7 and 8 for examples). These space constraints are larger than the amount of space used by the baseline versions of those applications but are reasonable for short-lived dialog boxes and our particular hardware configurations. We used the same space constraints for all participants to make results comparable.

Participants performed 6 sets of tasks with each of the interfaces. The first set counted as practice and was not used in the final analysis. Each set included between 9 and 11 operations, such as setting a widget's value or clicking a button; however, if a particular interface included tab panes, interactions with tab panes were recorded as additional operations. For example, if the user had to access Font Style after setting Text Effects in the baseline font formatting interface (Figure 5), they would have to perform two separate operations: click on the Font tab and then select the Style.

During each set of tasks, participants were guided visually through the interface by an animated rectangle shown in Figure 6. An orange border indicated what element was to be manipulated while the text on the white banner above described the action to be performed. As soon as the participant set a value of a widget or clicked on a tab, the rectangle animated smoothly to the next interface element to indicate the next task to be performed. The animation took 235 ms. We chose to use this approach because we were interested in studying physical efficiency of the candidate interfaces separate from any other properties that may affect their usability. The animated guide eliminated most of the visual search time required to find the next element, although participants still had to find the right value to select within some widgets.

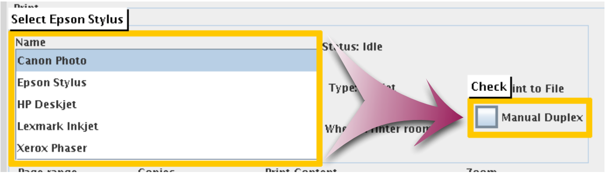


Figure 6. Participants were visually guided to the next element in the interface to be manipulated. The orange border animated smoothly to the next element as soon as the previous task was completed.

Procedure

We presented participants with the 9 interfaces: 3 applications (font formatting, print dialog, and synthetic) × 3 interface variants (baseline, preference-based, and ability-based) one at a time. Interface variants belonging to the same application were presented in contiguous groups. With each interface variant, participants performed 6 distinct task sets, the first being considered a practice (participants were told to pause and ask clarifying questions during practice task sets but to proceed at a consistent pace during the test sets). Participants were encouraged to take a break between task sets.

The tasks performed with each of the 3 interface variants for any of the 3 applications were identical and were presented in the same order. We counterbalanced the order of the interface variants both within each participant and across participants. The order of the applications was counterbalanced across participants.

After participants completed testing with each interface variant, we administered a short questionnaire, asking them to rate the variant's usability and aesthetics. After each block of three variants for an application, we additionally asked participants to rank the three interfaces on efficiency of use and overall preference. Finally, at the end of the study, we administered one more questionnaire recording information about participants' overall computer experience, typical computer input devices used, and their impairment (if any).

Generated Interfaces

Figure 7 shows three examples of user interfaces generated by SUPPLE++ based on participants' measured motor capabilities. These ability-based user interfaces tended to have widgets with enlarged clickable targets requiring minimal effort to set (e.g., lists and radio buttons instead of combo boxes or spinners). In contrast, user interfaces automatically generated by SUPPLE based on participants' stated preferences (see Figure 8) tended to be very diverse, as each participant had different assumptions about what interfaces would be easier to use for him or her.

Design and Analysis

The experiment was a mixed between- and within-subjects factorial design with the following factors and levels:

- *Impairment* {able-bodied (AB), motor-impaired (MI)}
- *Interface variant* {baseline, ability-based, preference-based}
- *Application* {font formatting, print dialog, synthetic}
- *Trial set* {1...5}
- *Participant* {1...17}

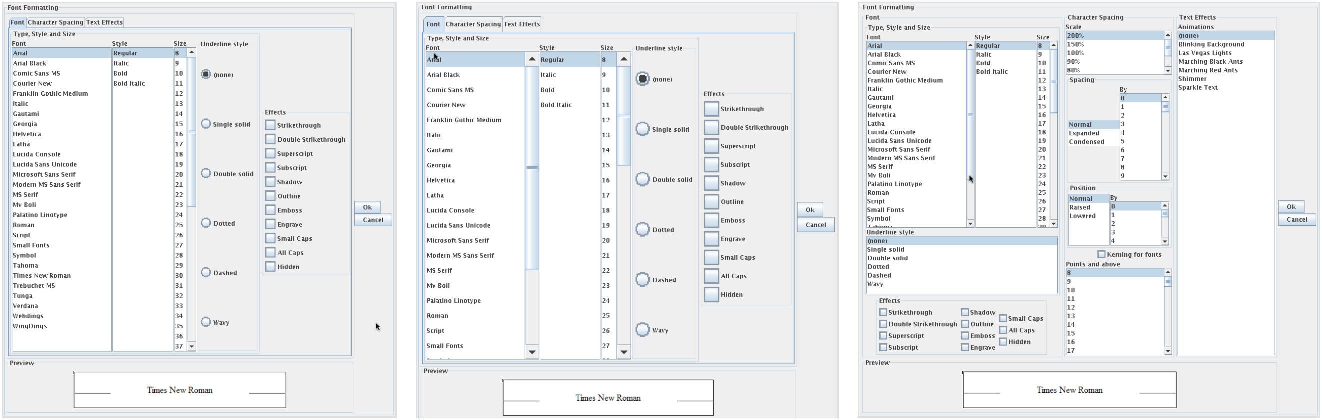
Participants completed $3 \times 3 \times 5 = 45$ trial sets each for a total of 765 trial sets (270 for able-bodied and 495 for motor-impaired).

Our dependent measures were:

- **Widget manipulation time** captures the time, summed over all operations in a trial set (including errors), spent by the participants manipulating individual widgets. It was measured from the moment of first interaction with a widget (first clicks or mouse wheel scroll in case of lists) to the moment the widget was set to the correct value. For many individual operations involving widgets like buttons, tabs, lists (if the target element was visible without scrolling), we recorded 0 manipulation time because the initial click was all that was necessary to operate the widget.
- **Interface navigation time** represents the time, summed over all operations in a trial set (including errors), participants spent moving the mouse pointer from one widget to the next; it was measured from the moment of effective start of the pointer movement to the start of the widget manipulation.
- **Total time** per trial set was calculated as a sum of widget manipulation and interface navigation times.
- **Error rate** per trial set was calculated as the fraction of operations in a set where at least one error was recorded; we regarded as "errors" any clicks that were not part of setting the target widget to the correct value.

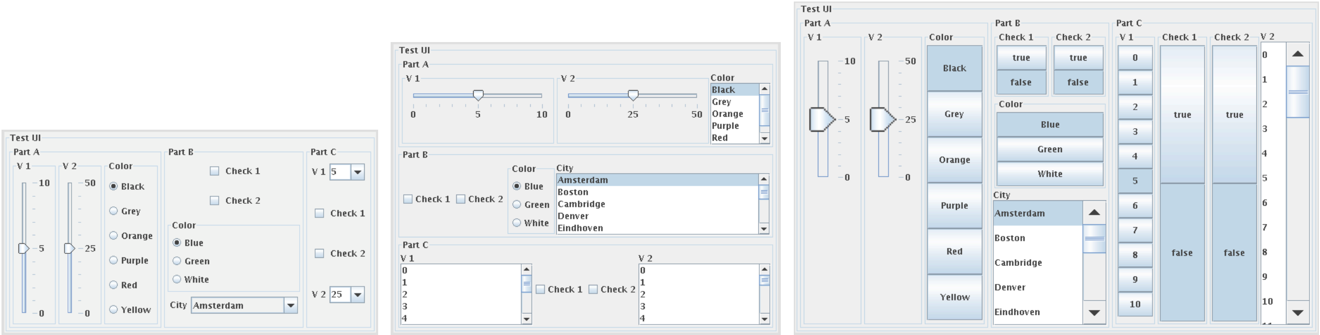
For each application and interface variant combination, we additionally collected 4 subjective measures on a Likert scale (1-7) relating to the interfaces' usability and attractiveness. We also asked the participants to rank order the 3 interface variants for each application by efficiency and overall preference.

For analysis, we took the logarithm of all timing data to adjust for non-normal distributions, which are often found in such data. We analyzed the timing data using a mixed-effects model analysis of variance with repeated measures: *Impairment*, *Interface variant*, *Application* and *Trial set* were modeled as fixed effects while *Participant* was modeled correctly as a random effect because the levels of this factor were drawn randomly from a larger population. Although such analyses retain larger denominator degrees of freedom, detecting statistical significance is no easier because wider confidence intervals are used [18, 25]. In our results, we omit reporting the effects of *Application* and *Trial set* because they were not designed to be isomorphic and naturally were expected to result in different performance. As often is the case, our error rate data was highly skewed towards 0 and did not permit analysis of variance. Accordingly, we analyzed error rates as count data using regression with an exponential distribution [27]. Subjective Likert scale responses were analyzed with ordinal logistic regression [29] and subjective ranking data with the Friedman non-parametric test.



AB02 MI02 MI04

Figure 7. User interfaces automatically generated by SUPPLE++ for the font formatting dialog based on three users' individual motor abilities. The interface generated for AB02 was typical for most able-bodied participants: small targets and tabs allow individual lists to be longer, often eliminating any need for scrolling (e.g., the font selection list). MI02 could perform rapid but inaccurate movements – all the interactors in this interface have relatively large targets (at least 30 pixels in either dimension) at the expense of having to use tab panes. In contrast, MI04 could move mouse slowly but accurately – this interface reduces the number of movements necessary by placing all the elements in a single pane at the expense of using smaller targets and lists that require more scrolling.



baseline AB03 MI09

Figure 8. User interfaces for the synthetic application. The baseline interface is shown in comparison to interfaces generated automatically by SUPPLE based on two participants' preferences. Able-bodied participants like AB03, preferred lists to combo boxes but preferred them to be short; all able-bodied participants also preferred default target sizes to larger ones. As was typical for many participants with motor-impairments, MI09 preferred lists to combo boxes and frequently preferred the lists to reveal a large number of items; MI09 also preferred buttons to either check boxes or radio buttons and liked larger target sizes.

Results

Adjustment of Data

We excluded 2/765 trial sets for two different motor-impaired participants: one due to an error in logging and one because the participant got distracted for an extended period of time by an unrelated event.

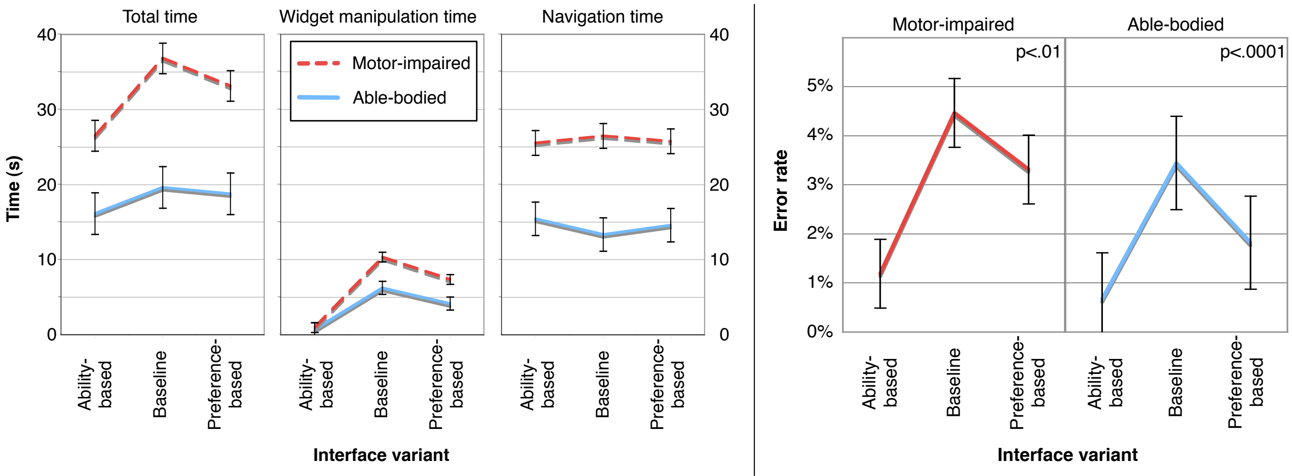
Completion Times

Both *Impairment* ($F_{1,15}$=28.14, $p < .0001$) and *Interface variant* ($F_{2,674}$=228.30, $p < .0001$) had a significant effect on the total task completion time. Motor-impaired users needed on average 32.2s to complete a trial set while able-bodied participants needed only 18.2s. The ability-based interfaces were fastest to use (21.3s), followed by preference-based (26.0s) and baselines (28.2s). A significant interaction between *Impairment* and *Interface variant* ($F_{2,674}$=6.44, $p < .01$) indicates that the two groups saw different gains over the baselines from the two personalized interface variants. Participants with motor-impairments saw significant gains: a 10% improvement for preference-based and a 28% improvement for ability-based interfaces ($F_{2,438}$=112.17, $p < .0001$). Able-bodied participants saw a relatively smaller, though still significant, benefit of the personalized interfaces: a 4% improvement for preference-based and 18% for ability-based interfaces ($F_{2,220}$=49.36, $p < .0001$).

The differences in performance can be explained by a significant[1] main effect of *Interface variant* on total manipulation time, that is, the time spent actually manipulating the widgets ($\chi^2_{(2,N=763)}$=359, $p < .0001$). With baseline interfaces, participants spent on average 8.29s per trial set manipulating the individual widgets. With preference-based interfaces, this number was 5.76s, while for ability-based interfaces, it was only 0.84s, constituting a nearly 90% reduction compared to baseline interfaces.

We also observed a significant main effect of *Interface variant* on the total navigation time ($F_{2,674}$=7.76, $p < .001$); baseline interfaces required the least amount of navigation

[1] The manipulation time data had bi-modal distribution because for many task sets the total manipulation time was 0. We therefore used a non-parametric Wilcoxon Rank Sum test [28] to analyze these data.



(a) Completion times

(b) Error rates

Figure 9. Participant performance. Both motor-impaired and able-bodied participants were fastest and made fewest errors with the ability-based interfaces. The baseline interfaces were slowest to use and resulted in most errors. Error bars show standard error.

time on average (19.9s) while preference- and ability-based interfaces required a little longer to navigate (20.2s and 20.5, respectively). While statistically significant, these differences were very small – on the order of 3% – and were offset by the much larger differences in total manipulation time. There was a significant interaction between *Impairment* and *Interface variant* with respect to the total navigation time ($F_{2,674}$=9.20, $p < .0001$): for able-bodied participants, navigation time was longer for both of the personalized interfaces ($F_{2,220}$=17.18, $p < .0001$), while for motor-impaired participants the effect was opposite, though smaller in magnitude and not significant.

Error Rates

We observed a significant main effect of *Interface variant* on the error rate ($\chi^2_{(5,N=153)}$=55.46, $p < .0001$): while the average error rate for baseline interfaces was 3.96%, it dropped to 2.57% for preference-based interfaces and to 0.93% for ability-based interfaces. This means that participants were both significantly faster *and* more accurate with the ability-based interfaces. There was no significant interaction between *Impairment* and *Interface variant* and the effects were similar and significant ($\chi^2_{(2,N=54)}$=23.66, $p < .0001$ for able-bodied and $\chi^2_{(2,N=99)}$=11.00, $p < .01$ for motor-impaired) for both groups individually (Figure 9).

Subjective Results

On a *Not Easy* (1) - *Easy* (7) scale for ease of use, motor-impaired participants rated ability-based interfaces easiest (6.00), preference-based next (5.64), and baseline most difficult (4.18). Similarly for able-bodied participants: 5.29 for ability-based, 5.00 preference-based and 4.38 for baseline. For both groups, these differences were significant ($\chi^2_{(2,N=99)}$=40.40, $p < .0001$ for motor-impaired, and $\chi^2_{(2,N=63)}$=6.95, $p < .05$ for able-bodied) and are summarized in Figure 10.

On a *Not Efficient* (1) - *Efficient* (7) scale, motor-impaired participants also found ability-based interfaces to be most efficient (5.58), followed by preference-based (5.18) and baseline interfaces (4.09). This difference was significant ($\chi^2_{(2,N=99)}$=23.31, $p < .0001$), but no corresponding significant difference was observed for able-bodied participants.

Similarly, on a *Not Tiring* (1) - *Tiring* (7) scale for how physically tiring the interfaces were, motor-impaired participants found baseline interfaces to be much more tiring (4.09) than either preference-based (3.12) or ability-based (2.61) variants ($\chi^2_{(2,N=99)}$=25.69, $p < .0001$), while able-bodied participants did not see the three interface variants as significantly different on this scale.

On a *Not Attractive* (1) - *Attractive* (7) scale for visual presentation, able-bodied participants found ability-based interfaces much less attractive (3.24) than either preference-based (4.90) or baseline variants (5.14). This effect was significant ($\chi^2_{(2,N=63)}$=25.52, $p < .0001$). Importantly, motor-impaired participants saw no significant difference in the attractiveness of the different interface variants.

When asked to rank order the three interface variants for each application by efficiency of use and overall preference (Table 3), both groups of participants ranked ability-based interfaces as most efficient, followed by preference-based, and then baseline interfaces. This result was only significant for participants with motor impairments ($\chi^2_{(2,N=33)}$=21.15, $p < .001$).

	Motor-impaired			Able-bodied		
	Ability-based	Baseline	Preference-based	Ability-based	Baseline	Preference-based
Efficiency	1.48	2.61	1.91	1.71	2.29	2.00
OverallRank	1.64	2.48	1.88	1.95	2.00	2.05

Table 3. Average subjective ranking by efficiency and overall preference (1=best, 3=worst)

With respect to overall preference, participants with motor impairments significantly preferred the two personalized types of interfaces than the baselines ($\chi^2_{(2,N=33)}$=12.61,

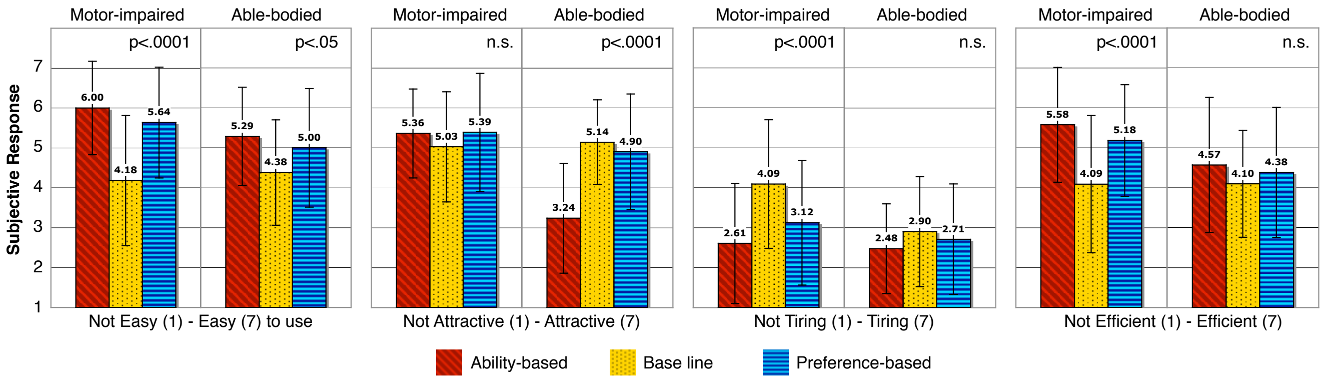


Figure 10. Subjective results. Both groups of participants found ability-based interfaces easiest to use. Motor-impaired participants also felt that they were most efficient and least tiring. Able-bodied participants found ability-based interfaces least attractive but, interestingly, motor-impaired participants saw little difference in attractiveness among the three interface variants. Error bars correspond to standard deviations. Note that on all graphs higher is better except for Not Tiring-Tiring.

$p < .01$). Able-bodied participants had no detectable preference for any of the interface variants.

Participant Comments

MI01, whose dexterity started to deteriorate only recently, commented that the baseline interfaces would be what she had preferred just a few years earlier, but now she found both kinds of the personalized interfaces preferable. MI02, who controls a trackball with his chin and types with a head-mounted wand, said that he uses the trackball only about 20% of the time when manipulating GUIs and the rest of the time he uses the keyboard because despite being slow, it is easier. If more interfaces were like the ability-based interfaces in the study, he would use the trackball more often.

MI06 observed that many widgets have pretty large clickable areas but that it is hard to tell that they are indeed clickable (e.g., labels next to radio buttons, white spaces after list items) and that clear visual feedback should be given when the mouse pointer enters such an area. Indeed, the impact of visual feedback on performance has been documented by others [1], and we also observed that many of our motor-impaired participants were very "risk-averse" in that they carefully moved the pointer to the center of the widget before clicking it, which they perhaps would not do if they could be sure that a click elsewhere would be effective.

DISCUSSION

Our participants with motor impairments were significantly faster, made many fewer errors, and strongly preferred automatically-generated personalized interfaces over the baselines. The results were particularly strong and consistent for ability-based interfaces adapted to their actual motor capabilities by SUPPLE++: participants were between 8.4% and 42.2% (mean=26.4%) faster with those interfaces than with the baselines, they preferred those interfaces to all others, and they found those interfaces the easiest to use, the most efficient, and least physically tiring. By helping improve their efficiency, SUPPLE++ helped narrow the gap between motor-impaired and able-bodied users by 62%, with individual gains ranging from 32% to 103%.

These results demonstrate that the current difference in performance between users with motor impairments and able-bodied users is at least partially due to user interfaces being designed with a particular set of assumptions in mind—assumptions that are inaccurate for users with motor impairments. By generating personalized interfaces which reflect these users' unique capabilities, we have shown that it is possible to greatly improve the speed and accuracy of users with motor impairments, even when they use standard input devices such as mice and trackballs.

Our results also confirm that the right trade-off in user interface design depends on a particular user's individual capabilities. Even able-bodied participants were faster and made fewer errors with ability-based interfaces, and even they recognized these interfaces as significantly easier to use than the alternatives. In the end, however, they found those interfaces—which exchanged sparseness and familiar aesthetics for improved ease of manipulation—to be uglier and generally no more preferable than the baselines.

Particularly striking in our study was the situation of MI02, who was 2.85 times slower than an average able-bodied participant using baseline interfaces, but only 1.99 times slower when using interfaces designed for his unique abilities. MI02 controls the trackball with his chin and types on a keyboard with a head-mounted wand; therefore, keyboard shortcuts are also inconvenient for him. Furthermore, his speech is significantly impaired so he cannot use speech recognition software. He works as an IT consultant so his livelihood is critically dependent on being able to interact with computers effectively. Currently, he has to compensate with perseverance and long hours for the mismatch between the current state of technology and his abilities. He was the slowest participant in our study, but with SUPPLE++ he was able to close nearly half the performance gap between himself and able-bodied participants using baseline interfaces.

FUTURE WORK

We see three exciting directions for continued research. First, a long-term field study should investigate whether users with motor impairments will actually adopt automatically-generated interfaces. Second, in order to make practical the deployment of applications with SUPPLE++-style interface personalization, we must extend an existing interface-builder design tool, such as one of the popular Eclipse plug-ins, so that it can generate the functional specification required as input for the automatic interface generator [8]. Finally, preference-based interfaces deserve further attention. Our participants performed preference-elicitation *out of context*, judging the appearance of unfamiliar interfaces without sense for their likely usage pattern. If, instead, they could interleave preference elicitation with actual interface *usage*, they might well have demonstrated larger performance improvements.

CONCLUSION

We evaluated two systems which automatically generate personalized interfaces given a model of the user. SUPPLE++ uses a model of the user's motor capabilities, which is constructed by its Ability Modeler from a set of one-time motor performance tests. SUPPLE, on the other hand, uses a model of the user's preferences, which is built by the ARNAULD elicitation system.

Our results show that participants with motor impairments were significantly faster, made fewer errors, and strongly preferred automatically-generated personalized interfaces over the baselines. These results were especially strong for the ability-based interfaces produced by SUPPLE++: the motor-impaired participants were between 8.4% and 42.2% (26.4% on average) faster with those interfaces than with the baselines, they preferred those interfaces to all others, and they found those interfaces the easiest to use, the most efficient, and least physically tiring. It appears that the gap in performance between users with motor impairments and able-bodied users is at least in part due to a mismatch between motor-impaired users' capabilities and the assumptions underlying the design of typical interfaces.

In order to provide equitable access to the growing number of tools and resources available through the computer, it is important that all users, in particular those with motor or other impairments, have the option of using interfaces other than the manufacturers' defaults. Due to the great diversity of abilities among users, manual creation of personalized interfaces is clearly not scalable. Our results demonstrate that automatic generation of ability-based interfaces is feasible, and that the resulting interfaces improve both performance and satisfaction of users with motor impairments.

Acknowledgments. This work was funded in part by Microsoft Graduate Research Fellowship, Intel Research, NSF grant IIS-0307906, ONR grant N00014-06-1-0147, DARPA project CALO through SRI grant number 03-000225, and the WRF / TJ Cable Professorship.

REFERENCES

1. Akamatsu, M., MacKenzie, I. S., and Hasbroucq, T. A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. *Ergonomics*, *38*, 4 (1995), 816–827.
2. Bergman, E. and Johnson, E. Towards Accessible Human-Computer Interaction. *Advances in Human-Computer Interaction*, *5*, 1.
3. Bigham, J. P., Kaminsky, R. S., Ladner, R. E., Danielsson, O. M., and Hempton, G. L. Webinsight:: making web images accessible. *Proc. Assets '06*. ACM Press, New York, NY, USA, 2006, 181–188.
4. Carter, S., Hurst, A., Mankoff, J., and Li, J. Dynamically adapting GUIs to diverse input devices. *Proc. Assets '06*. ACM Press, New York, NY, USA, 2006, 63–70.
5. Dawe, M. Desperately seeking simplicity: how young adults with cognitive disabilities and their families adopt assistive technologies. *Proc. CHI'06*. (2006), 1143–1152.
6. Fichten, C., Barile, M., Asuncion, J., and Fossey, M. What government, agencies, and organizations can do to improve access to computers for postsecondary students with disabilities: recommendations based on Canadian empirical data. *Int J Rehabil Res*, *23*, 3 (2000), 191–9.
7. Gajos, K., Christianson, D., Hoffmann, R., Shaked, T., Henning, K., Long, J. J., and Weld, D. S. Fast and robust interface generation for ubiquitous applications. *Proc. Ubicomp'05*. Tokyo, Japan, 2005.
8. Gajos, K. and Weld, D. S. Supple: automatically generating user interfaces. *Proc. IUI'04*. ACM Press, New York, NY, 2004, 93–100.
9. Gajos, K. and Weld, D. S. Preference elicitation for interface optimization. *Proc. UIST '05*. ACM Press, New York, NY, 2005.
10. Gajos, K. Z., Wobbrock, J. O., and Weld, D. S. Automatically generating user interfaces adapted to users' motor and vision capabilities. *Proc. UIST'07*. ACM Press, New York, NY, 2007.
11. Harada, S., Wobbrock, J. O., and Landay, J. A. Voicedraw: A voice-driven hands-free drawing application. *Proc. ASSETS'07*. ACM Press, 2007.
12. Hornof, A., Cavender, A., and Hoselton, R. Eyedraw: a system for drawing pictures with eye movements. *Proc. ASSETS'04*. ACM Press, New York, NY, 2004, 86–93.
13. Hwang, F., Keates, S., Langdon, P., and Clarkson, J. Mouse movements of motion-impaired users: a submovement analysis. *Proc. Assets '04*. ACM Press, New York, NY, USA, 2004, 102–109.
14. International Organization for Standardization. 9241-9 Ergonomic requirements for office work with visual display terminals (VDTs)-Part 9: Requirements for non-keyboard input devices (2000).
15. Keates, S., Langdon, P., Clarkson, J. P., and Robinson, P. User models and user physical capability. *User Modeling and User-Adapted Interaction*, *12*, 2 (2002), 139–169.
16. Koester, H. Abandonment of speech recognition by new users. *Proc. RESNA03*. Atlanta, Georgia, 2003.
17. Law, C., Sears, A., and Price, K. Issues in the categorization of disabilities for user testing. *Proc. HCII'05*. 2005.
18. Littell, R., MIlliken, G., Stroup, W., and Wolfinger, R. *SAS System for Mixed Models*. SAS Institute, Inc., Cary, NC, 1996.
19. Mankoff, J., Dey, A., Batra, U., and Moore, M. Web accessibility for low bandwidth input. *Proc. Assets '02*. ACM Press, New York, NY, USA, 2002, 17–24.
20. Nichols, J., Myers, B. A., Higgins, M., Hughes, J., Harris, T. K., Rosenfeld, R., and Pignol, M. Generating remote control interfaces for complex appliances. *Proc. UIST'02*. Paris, France, 2002.
21. Phillips, B. and Zhao, H. Predictors of assistive technology abandonment. *Assist Technol*, *5*, 1 (1993), 36–45.
22. Savidis, A. Dynamic software assembly for automatic deployment-oriented adaptation. *Electronic Notes in Theoretical Computer Science*, *127*, 3 (2005), 207–217.
23. Savidis, A., Antona, M., and Stephanidis, C. A decision-making specification language for verifiable user-interface adaptation logic. *International Journal of Software Engineering And Knowledge Engineering*, *15*, 6 (2005), 1063 – 1094.
24. Savidis, A. and Stephanidis, C. Inclusive development: Software engineering requirements for universally accessible interactions. *Interacting with Computers*, *18*, 1 (2006), 71–116.
25. Schuster, C. and Von Eye, A. The relationship of anova models with random effects and repeated measurement designs. *Journal of Adolescent Research*, *16*, 2 (2001), 205–220.
26. Stephanidis, C. User interfaces for all: New perspectives into human-computer interaction. In C. Stephanidis, ed., *User Interfaces for All*, Lawrence Erlbaum, 2001. 3–17.
27. Vermunt, J. K. *Log-linear Models for Event Histories*. Sage Publications, 1997.
28. Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*, 6 (1945), 80–83.
29. Winship, C. and Mare, R. D. Regression models with ordinal variables. *American Sociological Review*, *49*, 4 (1984), 512–525.