

# Evaluating accessibility by simulating the experiences of users with vision or motor impairments

J. Mankoff  
H. Fait  
R. Juang

User studies involving users with disabilities often incur greater financial cost and complexity than those involving general populations. Consequently, accessibility issues may not be identified during the earlier phases of software design, when designs are still malleable. Additionally, it can be difficult to create controlled studies with multiple groups of very similar subjects due to the extremely heterogeneous nature of the impact of many motor and visual disabilities. This paper examines the feasibility of simulating the interaction experiences of users with low vision or motor impairments. Based on empirical studies in the literature of the impact of these impairments on the experience of computer use, we have developed EASE (Evaluating Accessibility through Simulation of User Experience), a tool that can help developers identify disability-related usability problems early in the design process. EASE can also be used to allow fine-grained exploration of user capabilities that are difficult to control, such as achievable typing speed. We present a study of the use of word prediction software that illustrates the value of fine-grained control over typing speed and that also shows word prediction is most useful at typing speeds between 5 and 8 words per minute.

## INTRODUCTION

According to the U.S. Census Bureau, almost one in five Americans has a disability.<sup>1</sup> Additionally, the group of disabled individuals is the only minority group of which any American might become a member at any time. According to an analysis of the Census Bureau's Current Population Survey, only 9.9 percent of people with disabilities used the Internet in 1999, as compared to 38.1 percent of the general population.<sup>2</sup> Regardless of age, income, or

education, a gap still exists for those with disabilities. For example, although ability to physically use a computer has not been directly measured, the ability to grasp and manipulate small objects can

©Copyright 2005 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the Journal reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of the paper must be obtained from the Editor. 0018-8670/05/\$5.00 © 2005 IBM

serve as a proxy for this information. Data collected in late 1997 showed that 13.5 million individuals age fifteen and older had difficulty grasping small objects,<sup>3</sup> suggesting possible problems with mouse and keyboard use.

Accessible systems, that is, systems which can be used by users who are disabled as well as by the average user, are challenging to build. Like usability, accessibility is a goal that requires iteration, customer-centered design, and significant expertise to be done right. Unfortunately, user testing with special populations often requires greater effort, time, and monetary commitments on the part of designers, developers, and their companies than user testing with the general population.<sup>4</sup> Additionally, it may be difficult to find homogeneous groups of participants or to carefully control for certain factors, such as maximum achievable typing speed. In this paper, we present EASE (Evaluating Accessibility through Simulation of User Experience), a tool that is independent of any particular application or operating system, which simulates the interaction experiences of users with low vision, color blindness, and motor impairments. In particular, EASE can help solve both problems of cost and problems of homogeneity.

In the following section we discuss the goals we have designed EASE to address, as well as the ethical considerations of using simulation with participants who are not disabled to understand the experiences of persons with disabilities. We then present a brief review of related studies to place our work in context. This is followed by a description of EASE and a discussion of the basis for the input and output effects used by EASE to simulate the experiences of people with vision and motor impairments. We then describe a user study illustrating the use of EASE to explore the impact of fine-grained differences in achievable typing speeds on the usefulness of word prediction software. We conclude with a summary of our work and an indication of future research directions.

#### **EASE: INTENDED APPLICATIONS AND ETHICAL CONSIDERATIONS**

EASE has two main intended applications: (1) influencing software development in the early stages of design, and (2) allowing assessment of the impact of fine-grained degrees of difference in vision or motor impairments on the experience of using a

computer. We discuss these applications in more detail in the following subsections.

#### **Influencing the early stages of design**

If developers could obtain feedback about accessibility problems associated with their software during the early stages of design, they could address those problems before they became entrenched mistakes. Unfortunately, the best method for testing accessibility is a full-fledged user study,<sup>5</sup> a costly undertaking that typically requires a finished system, not an early design. Lower-cost techniques appropriate for the early stages of design have been explored in the Web domain, and one particularly promising approach involves simulating user experiences.<sup>5</sup> However, little support exists for testing arbitrary end-user applications or the desktop environment itself for accessibility with respect to low vision or motor impairments. By allowing developers to explore an interface under constraints similar to those experienced by people with disabilities, it should be possible to help discover potential problems early in the design process.

The goal of EASE is not to reproduce the experience of disability, but rather to provide application developers with a means of quickly getting a sense of software accessibility problems. As such, EASE is meant to be used in conjunction with informal methods meant to identify problems at the early stages of design, such as expert review and other techniques popular among interaction designers.<sup>6</sup>

There are further benefits that developers can gain from using a simulation tool like EASE. For example, EASE allows designers to experience what their users might experience every day while trying to interact with their computers. An understanding of the accessibility issues that users face gives designers an experiential knowledge from which to design in the future. This could be of particular benefit to designers who are relatively new to the concept of accessibility.

Another benefit of using a tool like EASE for early interface evaluation involves its impact on user studies. Because EASE is not intended as a replacement for user studies, it is assumed that, after early and iterative design testing, at some later point designers will have people with disabilities test the use of the system. By this time, developers will have already found and fixed a number of usability and

accessibility problems in their systems. It follows that their user studies can then focus on problems that would not have been uncovered had the original problems not been fixed. Using a tool like EASE offsets the chances of a disastrous user study, a result that can lead to abandonment of a design or, worse yet, abandonment of the goal of accessibility.

### Controlled fine-grained studies

The symptoms displayed by individuals with low vision or motor impairments are extremely heterogeneous, due in large part to the variety of ways in which such disabilities may be expressed. This heterogeneity makes it difficult to carefully control and assess the impact of these disabilities on the experience of using a computer. However, an understanding of such fine-grained differences can help to influence the design of accessible software and assistive technologies. For this reason, researchers have begun to develop models and to manipulate the parameters of those models to investigate the impact of specific performance differences.<sup>7,8</sup> These models are based on human data but are mathematical in nature.

EASE provides an alternative approach for controlling the fine-grained performance differences that could not easily be explored in a study of people with typically heterogeneous disabilities. First, EASE can be used to explore controlled performance differences in situations where models do not yet exist. Second, although modeling has typically been used instead of human participants in the past, we believe that modeling, when combined with simulation, can allow experimenters to obtain additional information with the help of nondisabled participants that modeling alone would not provide. As an example, we describe later in this paper a study of the use of word prediction software by individuals with motor impairments and show that simulation is an effective approach to obtaining results similar to those found through actual user studies with participants who had motor impairments.

### Ethical considerations

The choice to use typically nondisabled participants in place of participants with disabilities is a controversial one.<sup>9-11</sup> The two primary concerns involve ethical issues and the validity of the data obtained in such studies. For example, Bedrosian points out that “for some [people with disabilities], it [the opportunity to participate] was the first

opportunity that they had to express their opinions to people who were interested in listening . . . bypassing their opinions for the sake of convenience would be disrespectful to the population.”<sup>9</sup> In terms of validity, in the domain of augmentative and alternative communication, Bedrosian argues that in most cases it is not known if the differences between users with and without disabilities affect the experimental measurements.<sup>9</sup> However, Bedrosian and Higginbotham acknowledge that “when non-disabled individuals can be employed validly as research subjects, they can provide an economical alternative . . .”<sup>11</sup>

The EASE design is based on a survey of literature regarding the characteristics of computer use by people with different types of motor and vision impairments.<sup>12-19</sup> By grounding our simulation in this research, we hope to increase its validity.

We believe that in the early stages of design, when rapid iteration is important and full-fledged studies are too costly to implement, even approximate feedback on potential problems is useful. In this situation, the use of typically nondisabled participants, combined with simulation, becomes a viable alternative. Additionally, in situations where it is very difficult to control for the large individual differences in the impact of disabilities on the experience of using a computer, a tool like EASE can enable experimentation that would otherwise be prohibitively difficult or expensive to carry out.

### RELATED WORK

The idea of simulating disability as a way of increasing understanding is a controversial but potentially valuable approach to disability research.<sup>9-11</sup> The primary source of controversy is that the lack of context and depth in such studies results in caricatures of disability that, rather than providing an understanding of the experience of disability, lead to a sense that people with disabilities are helpless victims. Despite this concern, some scholars of disability find simulation to be a useful tool, if only to highlight the disconnection between a typically functioning person’s understanding of disability and the true experience of living with a disability.

In spite of this controversy, simulation has been applied to Web accessibility research in the past. For example, IBM’s aDesigner<sup>20</sup> renders a Web page so

that text or images that cannot be viewed by a person who is blind are impossible to see (covered in black). WebAIM (Web Accessibility In Mind)<sup>21</sup> at Utah State University provides scripted simulations with a number of capabilities, including those that illustrate screen reader use, those that demonstrate the use of screen-enlarging software for users with low vision, and those that simulate distractibility and the accompanying frustrations that a user with cognitive disabilities might experience.

Current techniques for finding accessibility problems are very costly,<sup>4</sup> and available tools and research in this domain are thus almost completely limited to the Web.<sup>5,20,22–26</sup> Our past work has shown that exploring Web pages with the help of a screen reader could help developers find accessibility problems with Web pages.<sup>5</sup> Although this approach to simulating user experience was limited to the Web and to people who were blind, it showed the promise of simulation as an approach for finding accessibility problems during the early stages of design.

EASE builds on this past work but is more general than aDesigner or WebAIM. EASE specifically includes a wider range of impairments and extends beyond the Web to include any application interface, including the user's desktop computer itself. We describe EASE in detail in the sections that follow.

### EASE: FEATURES AND IMPLEMENTATION

As noted, EASE is a tool for simulating the impact of different impairments on the experience of using a desktop computer. For example, the practical impact of a tremor on computer use may include difficulty in using a mouse to select targets and, depending on the severity of the tremor, difficulty selecting the correct key on the keyboard, or difficulty releasing a key before it starts to repeat.<sup>16</sup>

EASE can modify the input and output of an arbitrary computer desktop at the operating-system level (across all running applications) to reflect errors or disturbances in typical computer use caused by different potential combinations of motor and visual impairments. We will refer to these modifications as *I/O effects* throughout this paper.

The I/O effects supported by EASE were selected to simulate performance errors reported in several previous studies.<sup>12–19</sup> EASE partly or fully supports

three output effects: blurring, occlusion, and red-green color blindness; and five input effects: positioning errors, pressing the wrong key, pressing multiple keys, accidental key repeat, and target misses. (See **Table 1** for a summary.) All of the effects are configurable by modifying a text file that EASE loads at runtime.

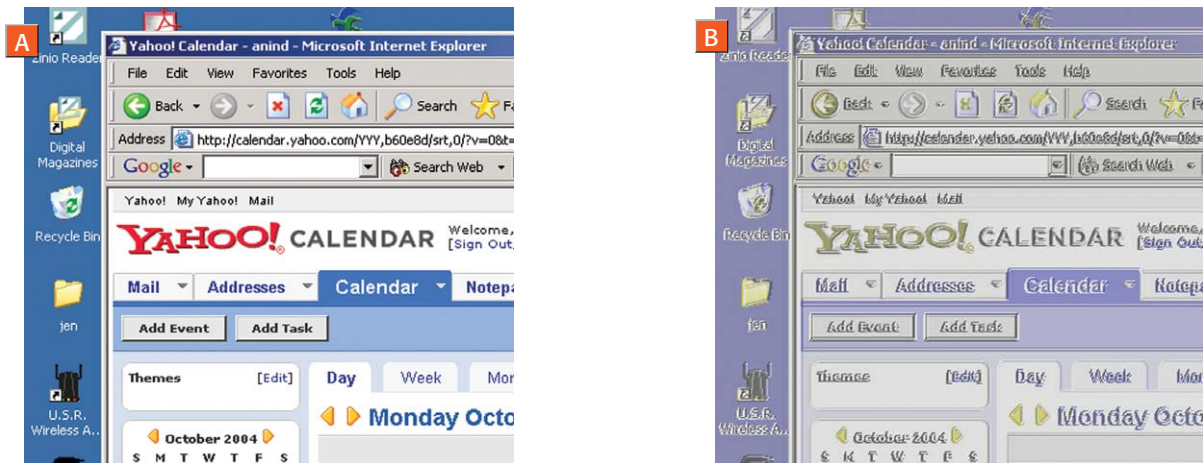
It should be noted that many assistive technologies are available that mitigate the impact of these input and output problems. For example, a variety of accommodations, such as assistive pointers<sup>27</sup> and magnification, can be used by people with low vision. Word prediction,<sup>7,28,32,34</sup> gravity wells,<sup>14</sup> assistive pointers,<sup>14</sup> scanning interfaces,<sup>8,33</sup> and other tools can assist people with motor impairments. EASE may be used in conjunction with such tools.

### Output effects

Users with low vision may experience lower visual acuity, lower contrast sensitivity, reduced field of vision, and reduced color perception.<sup>12</sup> In practice, specific common impairments such as cataracts, macular degeneration, and glaucoma each have characteristic effects on vision, as illustrated by WebAIM.<sup>13</sup> The specific output effects supported were based on these characteristic cases. Low vision can also have secondary effects on cursor use.<sup>29</sup> By simulating the primary effects of low vision, EASE can potentially create similar secondary effects. For this reason, we do not explicitly simulate those secondary effects. Currently, output effects supported by EASE include blurring and occluding different portions of the screen, as well as color modifications related to color blindness.

### Blurring

Blurring in EASE is implemented as a combination of factors. EASE modifies the screen to look slightly out of focus and causes the image to appear to be faded out. Fading is achieved by drawing a semi-transparent gray rectangle on top of the image of the desktop. The desktop is made to appear out of focus by convolving the desktop image using a matrix called a blurring kernel. The radius of the blurring-kernel matrix and the difference between blurred and unblurred areas are both parameterizable. This allows end users to control the degree to which the screen appears out of focus. **Figure 1A** shows an unblurred portion of a user's desktop, and **Figure 1B** shows the same desktop when blurred. As discussed



**Figure 1**  
A user's desktop: (A) unblurred; (B) blurred by EASE

by WebAIM,<sup>13</sup> cataracts have an effect similar to the combination of blurring and fading shown in Figure 1B.

### Occlusion

Several different types of low vision may result in occlusion of parts of the screen, including macular degeneration, glaucoma, and diabetic retinopathy.<sup>13</sup> Occlusion may obscure large portions of the screen either in the focal area of a person's vision (typical

of macular degeneration) or on the periphery (typical of glaucoma). Occlusion may also consist of many smaller artifacts scattered throughout both the focal area and the periphery (typical of diabetic retinopathy). Commonly, the edges of the occluded area are blurry, and, at the same time, areas that are not occluded may be completely clear.

Occlusion may be implemented by overlaying in black the occluded areas of the desktop and using

**Table 1** Input and output effects supported by EASE and their implementations. Each of the effects can be turned on individually, or effects can be combined.

Output effects	Blurring	Desktop image is faded and out of focus. Degree of focus is parameterizable.
	Occlusion	Oval is drawn at a parameterizable x/y offset near the cursor.
	Color modifications	Red and green are indistinguishable.
Input effects	Random mouse motion	With (parameterizable) random probability, each mouse motion event is perturbed a (parameterizable) random amount in the X and/or Y directions.
	Multiple key errors	With (parameterizable) random probability, a (parameterizable) number of adjacent keys are caused to generate events. Adjacency can be specified in a confusion matrix, with probability of selection specified for each element in the matrix. Effect is similar to literally pressing multiple adjacent keys on the keyboard.
	Repeat key errors	With (parameterizable) random probability, a key is repeated, also delaying the next keystroke for a (parameterizable) random amount of time.
	Incorrect key errors	A confusion matrix can be specified, with the probability that each of any number of other keys may be substituted for the key actually pressed.



transparency to create a fade effect at the edges of the occlusion. Without knowing exactly where the user is looking, it is difficult to perfectly mimic the effect of focal occlusion. However, the mouse cursor may be used as a rough approximation of a user's visual focus.

Unfortunately, completely blacking out the area under the mouse makes it extremely difficult to interact with an application because users cannot see anything under the mouse cursor. As an alternative, EASE allows an X and Y offset to be specified. This can be used to draw the occluded area in a position slightly offset from the cursor location. The user must then focus on the occluded area, using peripheral vision to guide the mouse, to explore the impact of this particular output effect.

EASE currently has partial support for occlusion. In the future, we plan to support arbitrary combinations of blurring and occlusion of multiple areas of the screen. This would allow EASE to represent most of the remaining output effects caused by different types of low vision, including macular degeneration, glaucoma, diabetic retinopathy, and cataracts.

### **Color modifications**

Color modifications can be used to make certain colors indistinguishable, as they are to people with color blindness. Color blindness occurs when the eye is not sensitive to certain wavelengths of light. In the most common cases, a person is not sensitive to either the long wavelengths (red colors) or medium wavelengths (green colors).

Color modifications are implemented in EASE by applying a filter to the desktop image. EASE modifies the red and green components of colors on the desktop image to be indistinguishable. In the future, we plan to allow the amount to which colors are conflated to be parameterizable.

### **Input effects**

The input effects supported by EASE are based on several different studies of how people with a variety of motor impairments use computers, as well as performance errors such individuals encounter.<sup>14-19</sup> The studies on which we based our work included participants with athetoid/ataxic cerebral palsy, spasm, Friedrich's ataxia, tremor, motor impairments caused by stroke, and Parkinson's disease, and also elderly users in general. These

studies included interviews and observations,<sup>15</sup> as well as detailed logging of mouse motion and keyboard use in various tasks.<sup>16-19</sup>

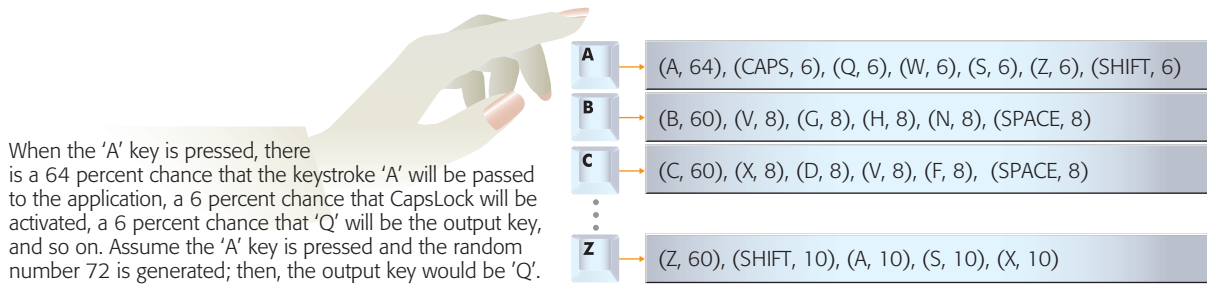
Input effects in EASE, including disturbances to both mouse and keyboard use, are under probabilistic control and are fully parameterizable. For example, it is straightforward to specify that when the user types 'f', it should be confused with 'd' or 'g' 5 percent of the time and with 'c', 'v', 'e' or 'r', each 1 percent of the time. This would simulate someone using a QWERTY keyboard, whose finger slips to one side or the other of the 'f' key as it is pressed, and far more rarely slips above or below that key.

### **Mouse motion**

The performance errors reported during mouse movement are quite varied. In order to describe such input errors, we must first describe a typical mouse motion. (See Reference 30 for a detailed discussion of research in motor learning and control.) A person moving a mouse to a target typically carries out a series of ballistic motions, meaning that the hand accelerates and decelerates smoothly in each segment of the series of motions. First, a user typically makes a large ballistic motion that moves the mouse pointer approximately to the target of interest. This is followed by a series of smaller, corrective motions. All of these motions typically are in the direction of the target, and each motion brings the mouse pointer closer to the target, until it is directly over the target. Finally, the user clicks or performs whatever action is needed.

The following are common errors observed in mouse movement:

1. Positioning errors<sup>16</sup> include additional submovements (additional motions in the series of ballistic motions leading to the target),<sup>17,18</sup> movement direction changes,<sup>17</sup> and indirect motion to a target.<sup>17</sup> Additionally, Trewin<sup>16</sup> found that users had more difficulty with positioning during dragging tasks and had difficulty keeping the mouse button down while dragging.
2. Target misses are caused by the mouse pointer entering and leaving a target multiple times<sup>17</sup> or moving during an attempted click.<sup>15,16,18</sup> The amount of motion may also increase near a target.<sup>18</sup>
3. Clicking errors include extended clicks, that is, holding the mouse button down too long and



**Figure 2**  
Error map used to determine the keystroke (output key) passed to an application after a participant presses a given input key

possibly moving the mouse off target while the button is down, as well as extended pauses between clicks, which would break up an intended double click.<sup>16</sup>

EASE supports the first two categories of errors, positioning errors and target misses, through a single mechanism that allows EASE to be configured to move the mouse pointer a random number of pixels (between a designated minimum and maximum) each time a mouse event occurs. This creates an effect similar to a tremor or spasm in the hand, forcing the user to add corrective submovements, and causes motion toward the target to be indirect. Additionally, the random motion effect may cause the mouse to overshoot a target during an attempted click. It is difficult to tell accurately when the mouse is over a target without knowing the user's intent, and thus EASE does not currently support increase in motion near targets. However, in the future, we plan to add support for increasing random motion as the ballistic submovements decrease in size, which typically only happens as the user nears a target.

**Keyboard use**

Trewin details the performance errors that have been measured during keyboard use,<sup>16</sup> including pressing multiple keys at once, holding a key down until key repeat starts by mistake, pressing the wrong key, pressing keys in the wrong order (i.e., transposing two keys), and difficulty in chording (holding one key down while pressing another).

In EASE, keyboard effects are all probabilistically controlled. For instance, the probability that multiple keys are pressed simultaneously can be specified in the dynamically loaded configuration file. Sim-

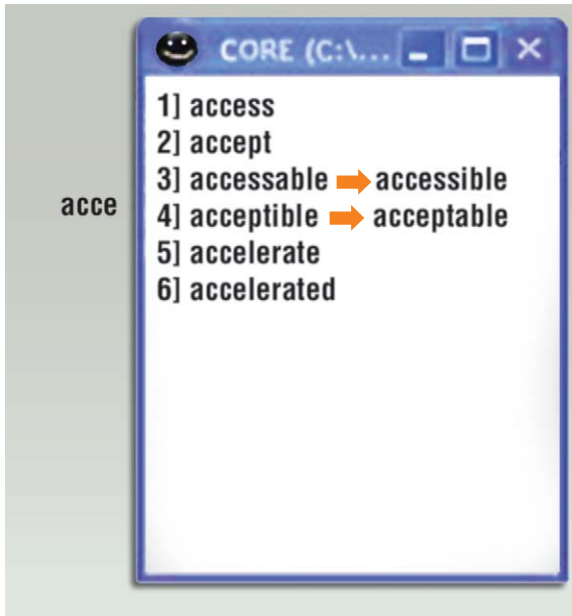
ilarly, the probability that a key will repeat (generate multiple characters when only pressed once) can be specified, along with the minimum and maximum number of times the key will repeat. EASE can also be configured to cause an incorrect key to appear instead of the key pressed by the user. This is done by specifying a confusion matrix for each key that might be mistaken for another. (A confusion matrix indicates which keys might be substituted for each possible key that is pressed, along with the probability of a substitution.) *Figure 2* shows an example of this effect.

**Implementation of EASE**

EASE is written in Java\*\* and is platform independent. It is implemented as a Virtual Network Computing (VNC) client and uses a modified version of the TightVNC client.<sup>31</sup> In particular, we modified the TightVNC methods responsible for handling input events from the client mouse and keyboard to create the intended input effects. We also modified the methods responsible for painting the remote desktop on the client screen to create output effects.

Because of the platform-independent nature of VNC, EASE can be used to connect to any VNC server running on any platform. It then displays the full desktop of the machine running the VNC server, including any applications, and gives the end user full control over that remote machine. Note that the VNC server and the EASE client must run on different machines.

Table 1 summarizes the specific features of EASE. Using a text file, the user can individually configure each of these features and turn them off or on. The



**Figure 3**  
Screen capture of the Aurora word prediction program

VNC client has been augmented with a button that can be used to load the configuration file, at which point the simulations go into effect.

### Summary of EASE

In summary, EASE is an application-independent simulation tool that can be used to modify the input and output of any machine that can run a VNC server. EASE simulations are based on studies of users with a variety of motor impairments, including those caused by athetoid/ataxic cerebral palsy, Friedrich's ataxia, stroke, Parkinson's disease, and aging. Although EASE does not simulate every aspect of the impact of these varied conditions on mouse, keyboard, and monitor use, it does include partial or full support for all of the major categories of errors. On the output side, this includes blurring, occlusion, and color blindness. On the input side, this includes positioning errors and target misses for mouse input, and key misses, pressing multiple keys, and accidental key repeat for keyboard input.

EASE was built on the assumption that simulating the types of performance errors encountered by users with motor and vision impairments can be helpful in testing accessibility issues in an interface or assistive technology. In the following sections, we describe one test of this assumption.

### DOES I/O SIMULATION WORK? A USER STUDY WITH EASE

Our user study focused one of the two primary applications of EASE described previously, namely, the ability to use simulation to control fine-grained differences in the impact of impairments on computer use. In particular, we tested the impact of different levels of achievable typing speeds on user performance with word prediction software. A word prediction application is an assistive technology that can increase the text production speed of users with motor impairments. More specifically, a word predictor is a software application running on a user's computer that displays to the user a list of words based on the letters that have already been typed. If the word predictor guesses correctly, the user can save keystrokes by selecting the correct word rather than typing the remaining letters. For instance, if a user wants to type the word "accessible," he or she normally has to type 10 characters. Using a word prediction tool, a user may type "a-c-c-e-s-i-b-l-e," checking the word prediction interface between each keystroke to see if the software has predicted the intended word. If the correct word appears after the user types "a-c-c-e," then the user chooses that word and has typed a total of five keystrokes (four letters and one selection key), or half of the total needed to type "accessible." *Figure 3* shows a screen capture of the word prediction tool that we used in our study.<sup>28</sup>

Word prediction systems are typically judged by a metric called keystroke savings, which measures the reduction in the number of keystrokes a user must type when using word prediction. (Thus, in our previous example the keystroke savings would be five characters.) Keystroke savings is a useful metric for comparison, but it is not a good way to measure the actual usefulness of word prediction to users because many other factors also affect the utility of word prediction. For instance, there is a high cognitive load associated with reading through possible words after each letter is typed, and it takes time to select the correct word once it appears.<sup>7</sup> Even at a savings of 50 percent, as in our example above, a user's overall input rate might not improve when using word prediction.

In practice, keystroke savings is only one component of successful word prediction use. By modeling fine-grained differences in word selection strategies, input capabilities, and other factors, researchers



have been able to explore these important features in more detail. For example, Leshner, et al. used modeling to explore the benefit of more accurate word prediction and compare it with many other aspects of scanning communication (single switch-based input).<sup>8</sup> They were able to show differences as large as 15 percentage points in the performance of different models. Koester and Levine used models to compare different strategies for using word prediction.<sup>7</sup> Using word prediction algorithms of different “quality” (low, average, and high keystroke savings), they compared strategies for deciding when to select words from a word list and when to keep typing. This was repeated for users with different typing speeds, as measured in keystroke times ranging from 0 to 2 seconds per key press, corresponding approximately to 0 to 6 words per minute (WPM). They were then able to recommend appropriate strategies for the most efficient use of word prediction in different situations. The many factors that the authors were able to explore helped to better illuminate the different trade-offs that affect the utility of word prediction for different users.

With the help of EASE, we were able to explore some of these issues without the overhead of first developing detailed models of how word prediction is used. Instead, we varied input speed and measured its effect empirically on user performance.

### Methodology

In testing users, our experimental conditions included four word-input speeds: 5, 8, 12, and 15 WPM. (Words average 5 characters, and thus 5 WPM is equivalent to 25 characters per minute, or approximately one character every two seconds.) We performed this study using a precursor to EASE that limited a participant’s input speed by accepting each new keystroke only after a time-out had occurred. Our tool buffered any keystrokes that occurred too early and waited for one time-out between each of them. Thus, if the wait time was 1 second and a user typed the word “Holly” without waiting between keystrokes, that user then had to wait 5 seconds for all five letters in “Holly” to appear on the screen.

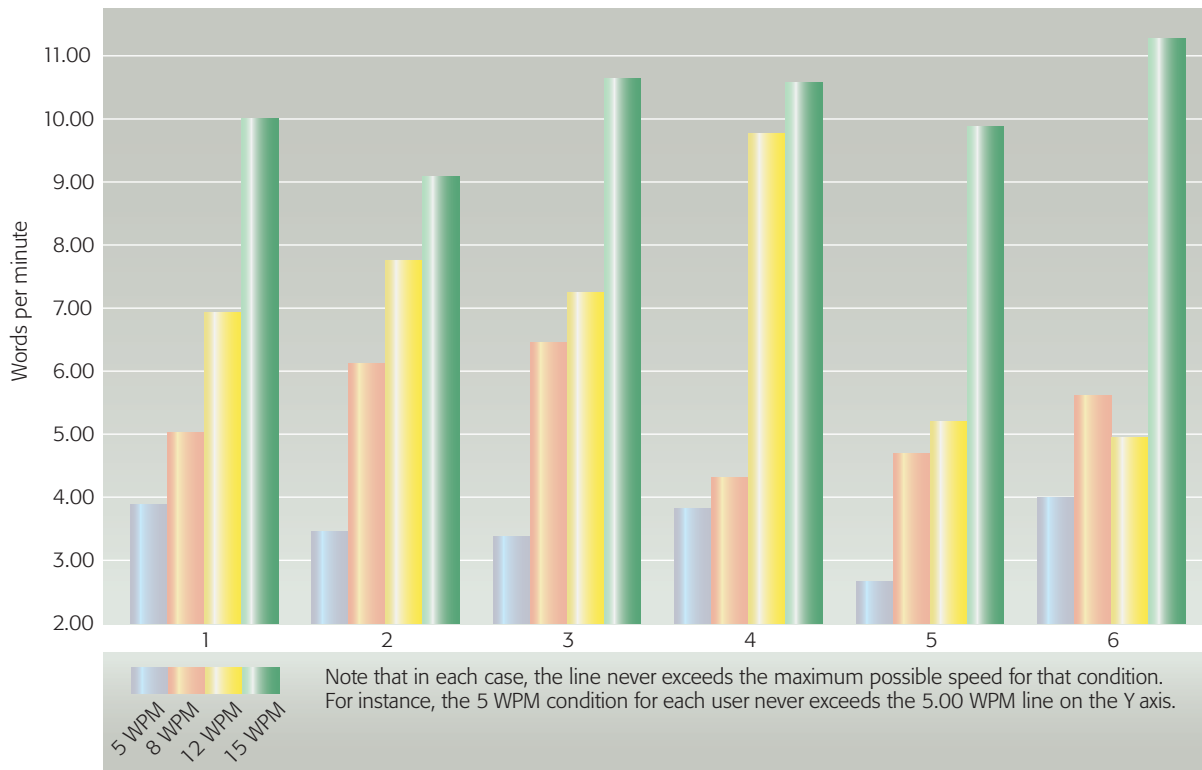
Our study included six participants with no known motor or cognitive impairments, all proficient at typing and comfortable using a computer for word-processing tasks. Participants were told to type as

accurately as possible but were not required to correct typing mistakes. The study was conducted using Microsoft Word for word processing and the Aurora software for word prediction (Figure 3). The Aurora tool is known to show an average keystroke savings of 46 percent on new texts.<sup>28</sup> Participants were given a tutorial on how to use the word prediction software and were encouraged to practice with it until they felt comfortable. Participants were not told when to use the word predictor and when to continue typing (i.e., participants were not told to “always read the word prediction list before typing another letter or choosing a word,” or “only read the word prediction list for words longer than three letters”). We asked the participants to respond, in random order, to four simple essay questions. A different maximum input rate (5, 8, 12, or 15 WPM) was set for each of these responses. Participants were instructed to wait until each letter appeared on the screen before typing another letter.

### Results

Participants became comfortable with EASE in less than 10 minutes. (However, despite our instructions, participants did not always wait until a letter appeared before typing the next letter.) All participants in all conditions produced text more slowly than the allowed WPM when using word prediction. *Figure 4* illustrates this graphically, showing the speed achieved by each user under each condition. Only the results for users typing at 5 WPM were inconclusive, because input rates with and without word prediction were very close. In fact, we found in a small separate study that with practice users could achieve speeds greater than 5 WPM when using word prediction.

At higher input speeds we found that the greater the maximum allowed input speed, the greater the difference between a participant’s actual speed and that maximum possible speed. These findings are illustrated graphically in *Figure 5*. For instance, when input is bounded at 12 WPM, we determined with 99 percent probability that participants will, on average, only achieve speeds of 9 WPM or less. Similarly, when input is bounded at 15 WPM, we determined with 99 percent probability that participants will, on average, only achieve speeds of 10 WPM or less. These calculations represent participants achieving only 75 percent and 66 percent of maximum speed, respectively. When input is bounded at 8 WPM, participants will on average



**Figure 4**  
Maximum input speeds achieved by the six study participants using the word prediction tool and varying input speeds

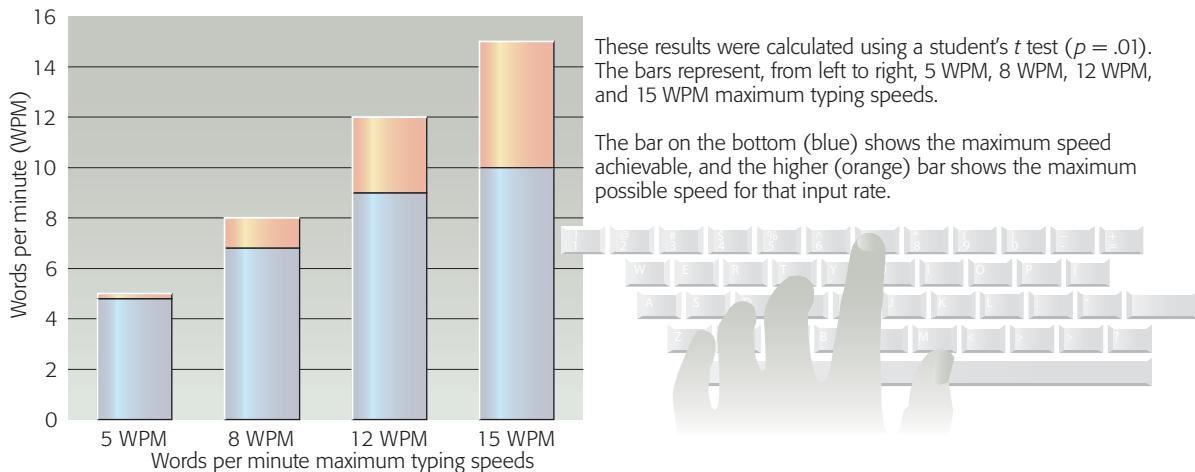
only achieve 6.8 WPM, or 85 percent of maximum speed. These findings show a clear drop in the ability of users to approach maximum possible speeds at higher input rates. Thus, the greater the input bandwidth of the user, the bigger the negative impact the participant experiences when using word prediction. When measuring the percentage of maximum speed each user achieved, no learning effects between speed conditions were found in the results.

Four of our six participants reported that not using word prediction, or ignoring the word prediction screen part of the time, allowed them to input text more quickly. This is possibly due to not having to split their attention between two tasks (composing a sentence and scanning a list). Participants also reported and were observed using the word prediction tool less effectively and only for longer words when typing at higher speeds. Also typical of users of word prediction, users in our simulation conditions were often observed typing an entire

word despite its availability on the word prediction list.

#### Discussion of results

As stated above, the percentage of maximum input rate achieved declined as input speeds increased. This result is interesting in that there are two factors in contention: (1) participants reported and were observed to be using word prediction less, and thus, they reported and were observed to be increasing their input rate, but (2), their input rates were still not at the maximum. However, initial tests suggest that this is not an effect of the input rate throttle because, in pilot testing of the throttle at 15 WPM, users could achieve a maximum rate of 14–15 WPM without the use of word prediction. Overall, when participants were allowed to type faster, they used the word prediction software less than when they were held to slower typing. Using simulation in this manner to test the word prediction tool leads us to conclude that there is a threshold at which users achieve no benefit, or are actually negatively



**Figure 5**  
Maximum typing speed achieved compared to maximum possible speed

affected, by word prediction technologies. Because participants, on average, achieved slower speeds with word prediction than without word prediction in all but the 5 WPM condition, we can conclude that this threshold is likely between 5 and 8 WPM.

We tested our work by comparing our results with those found through modeling or tests with participants who have motor impairments. If simulation is a valid approach to exploring fine-grained differences, then simulation should be able to achieve similar results to modeling and should not contradict the experiences of participants with motor impairments. In fact, our results are in agreement with observations in a study carried out by Koester and Simpson,<sup>32</sup> in which disabled participants with reduced input bandwidth caused by motor impairment were found to gain no benefit from using word prediction (without specialized configurations). However, it should be noted that there are many factors affecting performance with word prediction,<sup>7,33,34</sup> and typing speed is only one of these.

In summary, use of our tool for simulation of interaction experiences of users with motor impairments gave insights into the limitations of word prediction. In addition to showing feasibility in the domain of word prediction, this simulation was lightweight and required no special expertise. Learning to use our tool in conjunction with the word prediction software took our participants less than 10 minutes, largely because EASE is used with a standard keyboard, mouse, and monitor. Although

we did not validate the application of EASE to testing systems during early-stage iterative design, the low learning curve is supportive of that possibility.

One implication of these findings is that developers might find themselves tempted to use simulation as a replacement for user testing or as an excuse to rely on intuition. Such an approach is entirely contrary to the purpose of our method. Our method is designed to give developers a broader understanding of what it means to rely on unimpaired accessibility and to provide developers with a lightweight, easy-to-learn, easy-to-use method to evaluate the accessibility of their applications in early design stages. EASE is not an appropriate replacement for the essential information that users contribute.

## CONCLUSIONS

This paper presents EASE, a tool that can be used to model the impact of a variety of physical impairments to the experience of using a computer. EASE focuses specifically on the measurable effects of different impairments on keyboard and mouse use and on the appearance of output on a monitor.

EASE works at the operating-system level and can be used with any running application on any system supporting VNC. It allows developers to explore the accessibility of an application themselves or to observe others as they use an application. EASE is fully configurable at runtime, and any or all input

and output effects can be applied separately or in combination.

EASE has two major applications. One is controlled evaluation of the impact of differences in performance on accessibility and assistive technologies. We explored this use of EASE in a study testing the impact of different typing speed limits on the use of word prediction software, and we found that users derive no benefit from word prediction when they can type at or above 8 words per minute.

A second application of EASE is helping developers find accessibility problems during early stages of design. The low learning curve for EASE, combined with past work studying lightweight approaches to finding usability problems, provides compelling preliminary support for the applicability of EASE in early design. In the future, we plan to further explore this use of EASE and to test appropriate methodologies for using EASE during design. Additionally, we plan to explore other applications of EASE, such as training developers and other interested individuals on the subject of accessibility.

We also plan to validate the accuracy of EASE as a simulation tool. The EASE design is based on existing studies of the specific impacts of low-vision and motor impairments on the experience of using a computer mouse, keyboard, and monitor. There are many open questions in this area, and it is unlikely that the EASE I/O effects will ever be identical to those experienced by the users it is trying to simulate. However, it is important to empirically validate the fact that the EASE features affect interaction with software in a way that is comparable to the experience of persons with low vision or motor impairments.

\*\*Trademark, service mark, or registered trademark of Sun Microsystems, Inc.

## CITED REFERENCES

1. Facts and Features: CB02-FF.11, U.S. Census Bureau (July 12, 2002), <http://www.census.gov/Press-Release/www/2002/cb02ff11.html>.
2. H. S. Kaye, *Computer and Internet Use Among People with Disabilities*, Disability Statistics Center, Institute for Health and Aging, University of California, San Francisco (March 2000), [http://dsc.ucsf.edu/view\\_pdf.php?pdf\\_id=23](http://dsc.ucsf.edu/view_pdf.php?pdf_id=23).
3. *Americans with Disabilities: Household Economic Studies*, U.S. Census Bureau (February 1997), <http://www.census.gov/prod/2001pubs/p70-73.pdf>.

4. K. P. Coyne and J. Nielsen, *How to Conduct Usability Evaluations for Accessibility: Methodology Guidelines For Testing Websites and Intranets with Users Who Use Assistive Technology*, Nielsen Norman Group (October 2001), <http://www.nngroup.com/reports/accessibility/testing/>.
5. J. Mankoff, H. Fait, and T. Tran, "Is Your Web Page Accessible? A Comparative Study of Methods for Assessing Web Page Accessibility for the Blind," *Proceedings of the 2005 Conference on Human Factors in Computing Systems (CHI 2005)*, Portland, OR, April 2-7, 2005, ACM, New York (2005), pp. 41-50.
6. K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey, "A Survey of User-Centered Design Practice," *Proceedings of the 2002 Conference on Human Factors in Computing Systems (CHI 2002)*, Minneapolis, MN, April 20-25, 2002, ACM, New York (2002), pp. 471-478.
7. H. H. Koester and S. P. Levine, "Model Simulations of User Performance with Word Prediction," *Augmentative and Alternative Communication* **14**, No. 1, 25-35 (March 1998).
8. B. J. Moulton, G. W. Leshner, and D. J. Higginbotham, "Techniques for Augmenting Scanning Communication," *Augmentative and Alternative Communication* **14**, 81-101 (1998).
9. J. L. Bedrosian, "Limitations in the Use of Non-Disabled Subjects in AAC Research," *Augmentative and Alternative Communication* **11**, 6-10, (March 1995).
10. D. J. Higginbotham, "The Use of Non-Disabled subjects in AAC Research: Confessions of a Research Infidel," *Augmentative and Alternative Communication* **11**, 2-5, (March 1995).
11. D. J. Higginbotham and J. Bedrosian, "Subject Selection in AAC Research: Decision Points," *Augmentative and Alternative Communication* **11**, 11-13, (March 1995).
12. J. A. Jacko, M. A. Dixon, R. Rosa, I. U. Scott, and C. J. Pappas, "Visual Profiles: A Critical Component of Universal Access," *Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI 1999)*, Pittsburgh, PA, May 15-20, 1999, ACM, New York (1999), pp. 330-337.
13. Visual Disabilities: Low Vision, WebAIM, Center for Persons with Disabilities, Utah State University, <http://www.webaim.org/techniques/visual/lowvision>.
14. A. Worden, N. Walker, K. Bharat, and S. Hudson, "Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons," *Proceedings of the 1997 Conference on Human Factors in Computing Systems (CHI 1997)*, Atlanta, GA, March 22-27, 1997, ACM, New York (1997), pp. 266-271.
15. M. Y. Ivory, J. Mankoff, and A. Le, "Using Automated Tools to Improve Web Site Usage by Users with Diverse Abilities," *Information Technology and Society* **3**, No. 1, 195-236 (2003).
16. S. Trewin, "A Study of Input Device Manipulation Difficulties," *Proceedings of the Second International ACM SIGCAPH Conference on Assistive Technologies (ASSETS 1996)*, Vancouver, BC, Canada, April 11-12, 1996, ACM, New York (1996), pp. 15-22.
17. S. Keates, F. Hwang, P. Langdon, P. J. Clarkson, and P. Robinson, "Cursor Measures for Motion-Impaired Computer Users," *Proceedings of the Fifth International ACM SIGCAPH Conference on Assistive Technologies (ASSETS 2002)*, Edinburgh, Scotland, UK, July 8-10, 2002, ACM, New York (2002), pp. 135-142.

18. F. Hwang, "A Study of Cursor Trajectories of Motion-Impaired users," *Extended Abstracts of the 2002 Conference on Human Factors in Computing Systems (CHI 2002)*, Minneapolis, MN, April 20-25, 2002, ACM, New York (2005), pp. 842-843.
19. N. Walker, D. A. Philbin, and A. D. Fisk, "Age-Related Differences in Movement Control: Adjusting Submovement Structure to Optimize Performance," *The Journals of Gerontology, Series B: Psychological Sciences and Social Sciences* **52**, No. 1, 40-52 (January 1997).
20. P. Dvorak, "Looking for a User-Friendly Internet," *The Wall Street Journal*, September 16, 2004.
21. WebAIM Simulations, WebAIM, Center for Persons with Disabilities, Utah State University, <http://www.webaim.org/simulations/>.
22. Bobby™: *Testing Web Content Accessibility*, Watchfire Corporation, <http://www.watchfire.com/products/desktop/bobby>.
23. LIFT, UsableNet Inc., <http://www.usablenet.com/>.
24. W3C® Markup Validation Service, World Wide Web Consortium (W3C), <http://validator.w3.org/>.
25. M. Rowan, P. Gregor, D. Sloan, and P. Booth, "Evaluating Web Resources for Disability Access," *Proceedings of the Fourth International ACM SIGCAPH Conference on Assistive Technologies (ASSETS 2000)*, Arlington, VA, November 13-15, 2000, ACM, New York (2000), pp. 80-84.
26. D. Sloan, P. Gregor, M. Rowan, and P. Booth, "Accessible Accessibility," *Proceedings of the ACM Conference on Universal Usability (CUU 2000)*, Washington, DC, November 16-17, 2000, ACM, New York (2000), pp. 96-101.
27. J. Fraser and C. Gutwin, "A Framework of Assistive Pointers for Low Vision Users," *Proceedings of the Fourth International ACM SIGCAPH Conference on Assistive Technologies (ASSETS 2000)*, Arlington, VA, November 13-15, 2000, ACM, New York (2000), pp. 9-16.
28. Aurora Systems, Inc., <http://www.aurora-systems.com/pages/index.html>.
29. J. A. Jacko, A. B. Barreto, G. J. Marmet, J. Y. M. Chu, H. S. Bausch, I. U. Scott, and R. H. Rosa Jr., "Low Vision: The Role of Visual Acuity in the Efficiency of Cursor Movement," *Proceedings of the Fourth International ACM SIGCAPH Conference on Assistive Technologies (ASSETS 2000)*, Arlington, VA, November 13-15, 2000, ACM, New York (2000), pp. 1-8.
30. *Advances in Motor Learning and Control*, H. N. Zelaznik, Editor, Human Kinetics Publishers, Inc., Champaign, IL (1996).
31. TightVNC Software, The TightVNC Project, <http://www.tightvnc.com/>.
32. H. H. Koester and R. C. Simpson, "Effect of System Configuration on User Performance with Word Prediction: Results for Users with Disabilities," *Proceedings of the RESNA 2000 Annual Conference*, June 28-July 2, 2000, Orlando, FL, RESNA Press, Arlington, VA (2000), <http://www.resna.org/ProfResources/Publications/Proceedings/2000/AugCom/WPDisabil.php>.
33. H. H. Koester and S. P. Levine, "Learning and Performance of Able-Bodied Individuals Using Scanning Systems with and without Word Prediction," *Assistive Technology* **6**, No. 1, 42-53 (1994).
34. H. H. Koester and S. Levine, "Effect of a Word Prediction Feature on User Performance," *Augmentative and Alternative Communication* **12**, No. 3, 155-168 (1996).

*Accepted for publication January 19, 2005.  
Published online August 4, 2005.*

#### **Jennifer Mankoff**

*Human Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3891 (jmankoff@cs.cmu.edu).* Dr. Mankoff is an assistant professor in the Human Computer Interaction Institute at Carnegie Mellon University. She earned a B.A. degree at Oberlin College and a Ph.D. degree in computer science at the Georgia Institute of Technology. Her research focuses on tools and techniques for rapid, iterative prototyping of ubiquitous computing applications and accessible technology. Application areas of her work include assistive technology for people with special needs as well as health and safety.

#### **Holly Fait**

*The Exploratorium, 3601 Lyon Street, San Francisco, CA 94123 (hfait@chilaquiles.org).* Ms. Fait is a researcher at The Exploratorium, The Museum of Science, Art and Human Perception. Her current research includes developing pedagogical meta-data for a digital library of exhibit-based resources for science teachers, evaluating visitor use of radio frequency identification (RFID) to extend museum visits, and online audience research. Her previous work in human computer interaction included a focus on assistive technologies for people with disabilities. She received a B.A. degree from Mills College, and an M.S. degree in computer science from the University of California, Berkeley.

#### **Ray Juang**

*11227 Avalanche Way, #B, Columbia, MD 21044 (radj3@yahoo.com).* Mr. Juang is a software engineer at the Department of Defense. His current work involves research in network and communications security. His previous research work was in human-computer interaction and machine vision. He received a B.S. degree in electrical and computer engineering from the University of California, Berkeley. ■