DESIGN USE BUILD

ACE Lab

Twenty-Five Years, Five Projects, and Five Reflections on Accessible Computing



Jacob O. Wobbrock

The Information School

University of Washington

Seattle, WA, USA

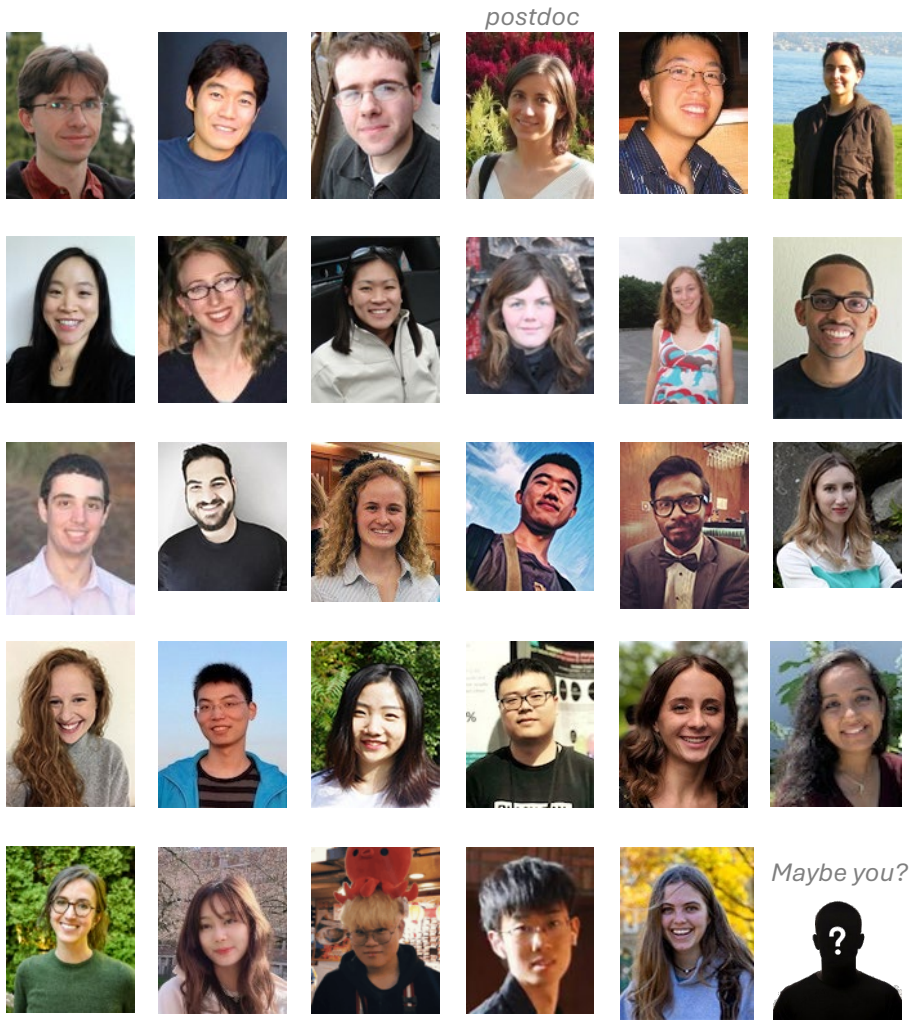
Information School
UNIVERSITY of WASHINGTON

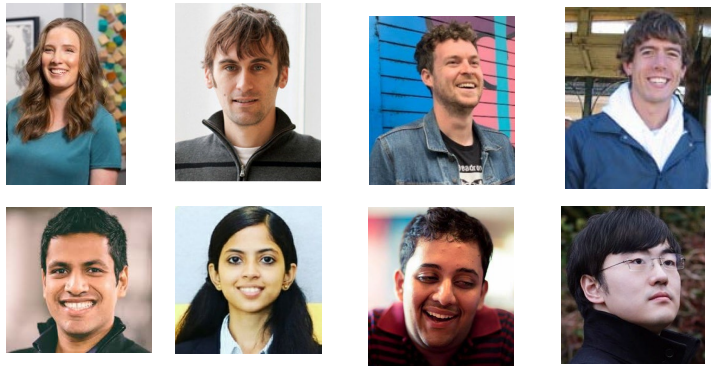
PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

CENTER FOR RESEARCH AND EDUCATION ON ACCESSIBLE TECHNOLOGY AND EXPERIENCES
UNIVERSITY of WASHINGTON

Acknowledgements

Information and Computer Science Ph.D. Students





ACE Lab

Acknowledgements

Faculty and Industry Collaborators

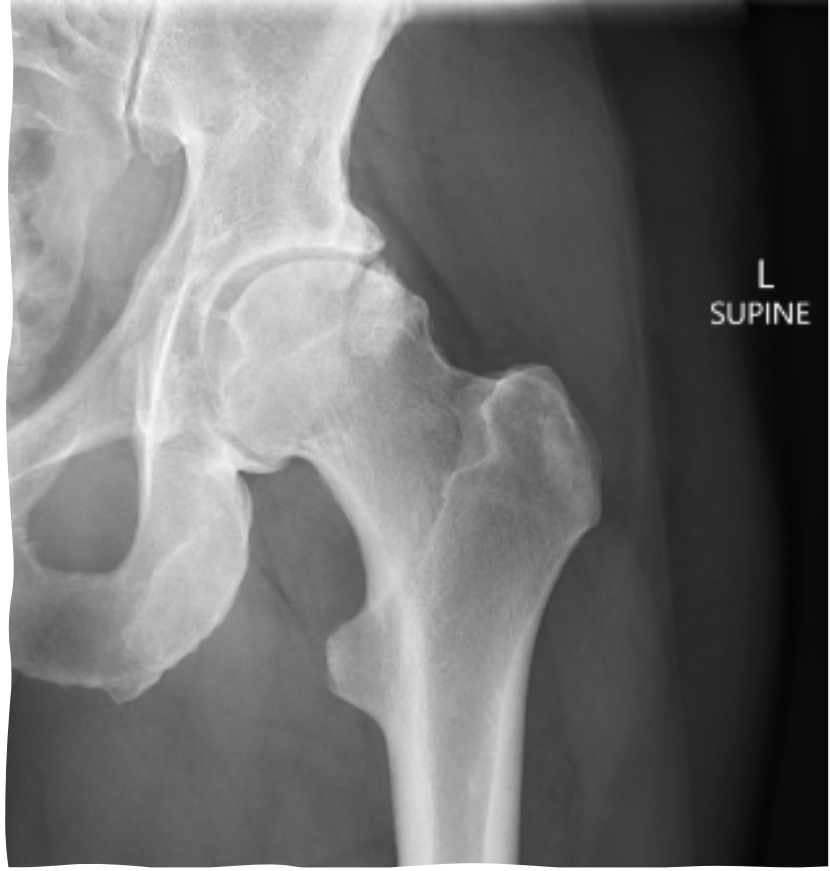


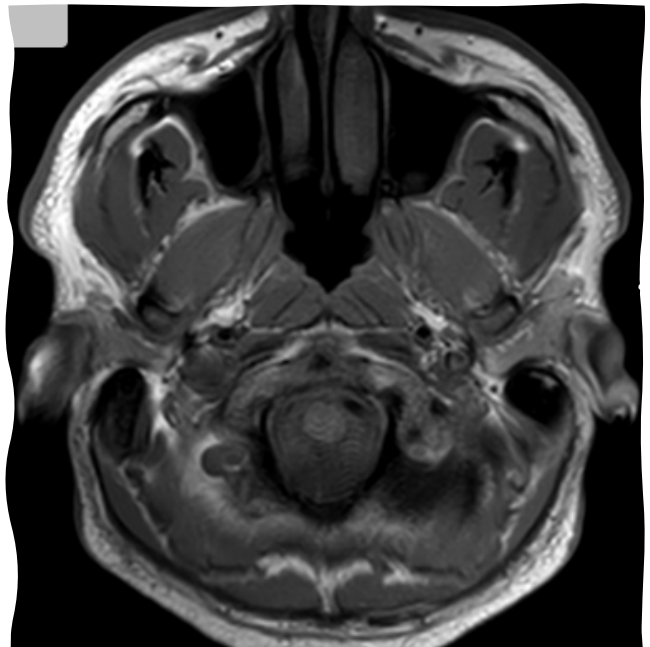


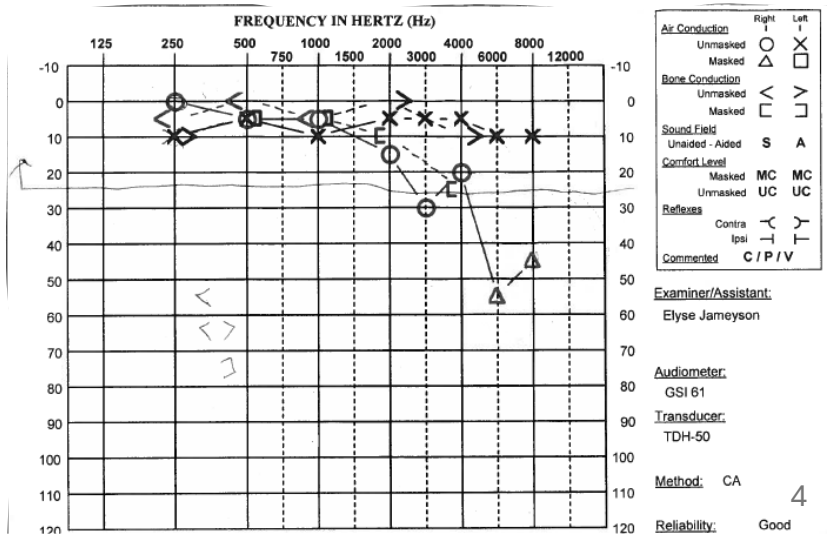
My relationship to accessibility…

- Hearing loss
- Chronic back pain
- Osteoarthritis
- Migraines
- Memory loss
- I have abilities and limitations, like anyone else
- Aging means accessibility is for everyone

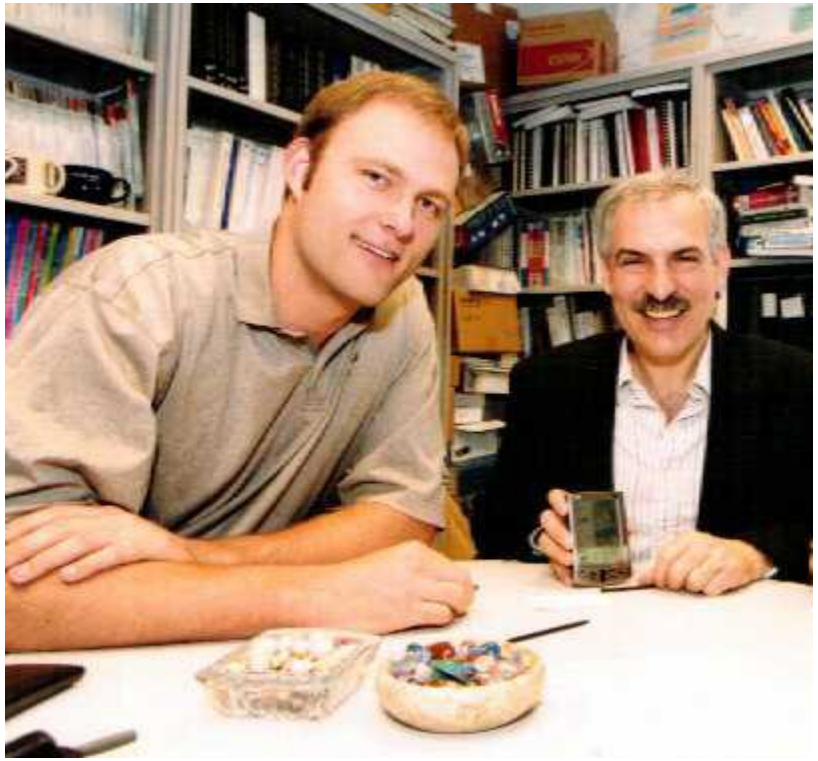








Brad Myers, my advisor





Using Handhelds to Help People with Motor Impairments

Brad A. Myers, Jacob O. Wobbrock, Sunny Yang, Brian Yeung, Jeffrey Nichols, Robert Miller
Human Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
bam@cs.cmu.edu
http://www.cs.cmu.edu/~pebbles/

ABSTRACT
People with Muscular Dystrophy (MD) and certain other muscular and nervous system disorders lose their gross motor control while retaining fine motor control. The result is that they lose the ability to move their wrists and arms, and therefore their ability to operate a mouse and keyboard. However, they can often still use their fingers to control a pencil or stylus, and thus can use a handheld computer such as a Palm. We have developed software that allows the handheld to substitute for the mouse and keyboard of a PC, and tested it with four people (ages 10, 12, 27 and 53) with MD. The 12-year old had lost the ability to use a mouse and keyboard, but with our software, he was able to use the Palm to access email, the web and computer games. The 27-year-old reported that he found the Palm so much better that he was using it full-time instead of a keyboard and mouse. The other two subjects said that our software was much less tiring than using the conventional input devices, and enabled them to use computers for longer periods. We report the results of these case studies, and the adaptations made to our software for people with disabilities.

Keywords: Assistive Technologies, Personal Digital Assistants (PDAs), Handicapped, Disabilities, Hand-held computers, Palm pilot, Muscular Dystrophy, Pebbles.

INTRODUCTION
About 250,000 people in the United States have Muscular Dystrophy (MD), which is the name given to a group of noncontagious genetic disorders where the voluntary muscles that control movement progressively degenerate. One form, called Duchenne Muscular Dystrophy (DMD), affects about one in every 4,000 newborn boys. With Duchenne, boys start to be affected between the ages of 2 and 6, and all voluntary muscles are eventually affected [10]. First affected are the muscles close to the trunk, and nearly all children with DMD lose the ability to walk sometime between ages 7 and 12. In the teen years, activities involving the arms, legs or trunk require assistance. Becker Muscular Dystrophy is a much milder version, and the onset can be in late adulthood.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Assets 2002, July 8-10, 2002, Edinburgh, Scotland
Copyright 2002 ACM ISBN 1-58113-464-9-02/07 …$5.00



Figure 1. 12-year old Kevin has Duchenne Muscular Dystrophy. He is operating his PC by using two hands to control the stylus on a Palm running Remote Commander.

A related disorder is Spinal Muscular Atrophy, which is an inherited neuromuscular disease that causes weakness in the body, arms, and legs. It affects both boys and girls starting at 6 months to 3 years, and progresses rapidly [10].

People with these and related disorders, like the rest of the population, are increasingly using computers. Unfortunately, these disorders often make it difficult for them to move their arms, wrists and fingers, and therefore conventional keyboards and mice become difficult or impossible for them to use. We are investigating how a handheld computer, such as the Palm, can be used to enable people with disabilities to access their PCs (see Figure 1).

This paper reports on how we have adapted our Pebbles software for the Palm to facilitate its use by people with motor impairments. In particular, we adapted the Pebbles Remote Commander program, which allows the Palm to be used as if it was the PC's keyboard and mouse. In general, the Pebbles project is studying many ways that handheld computers can interoperate with PCs, with each other, and with smart appliances [11]. These applications are aimed at business meetings, offices, classrooms, military command posts, and homes. The Pebbles software is available for downloading from: http://www.pebbles.hcii.cmu.edu/. Pebbles stands for **P**DAs for the **E**ntry of **B**oth **B**ytes and **L**ocations from **E**xternal **S**ources.

In a preliminary case study of four people with MD, we found that our modified Pebbles applications successfully allowed the use of the PC for extended periods of time for

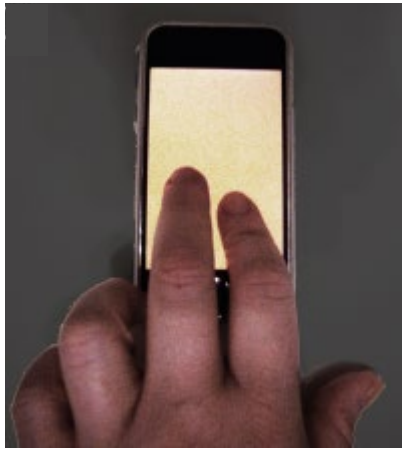

A.I. Generated
What would I tell
my younger self?

Five reflections



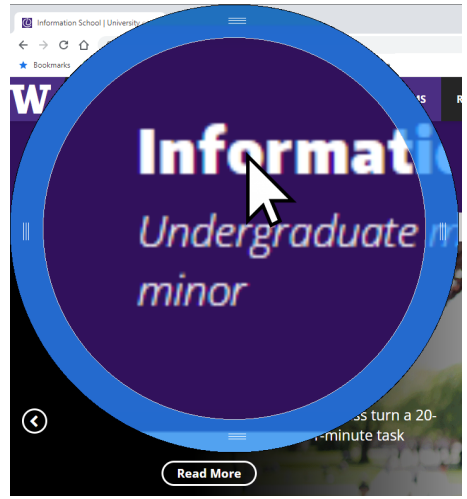
1. Prioritize depth

   EdgeWrite (2001 – 2006)



2. Participant-driven prototyping

   Slide Rule (2007 – 2008)



3. Deploy and support in the real world

   Pointing Magnifier (2010 – present)


contributed articles

DOI:10.1145/3148051

By focusing on users' abilities rather than disabilities, designers can create interactive systems better matched to those abilities.

BY JACOB O. WOBBROCK, KRZYSZTOF Z. GAJOS, SHAUN K. KANE, AND GREGG C. VANDERHEIDEN

Ability-Based Design

RECALL THE LAST time you took a trip out of town. Perhaps you were traveling to a conference far from home. Remember the many forms of transportation you endured: cars, buses, airplanes, and trains. Not only were you responsible for moving yourself over a great distance, you had to move your things as well, including books and baggage. Remember the cramped spaces, sharp elbows, body aches, and exhaustion. Feel again your desire to simply be at your destination with your possessions intact . . .

Such journeys remind us of our physical embodiment in the physical world, that much of our lived experience is fundamentally physical, and that we must contend with the world on physical terms. As computing professionals, we might be tempted to forget this, as our keystrokes summon data instantly from across the globe. But as humans, we still interact with that data through physical devices and displays using our physical senses and bodies. We and the world interact physically.

Civilization's story of technological progress is in no small part the story of an increasingly built physical environment, from the pyramids to roads to skyscrapers to sanitation systems. Much of our energy, collectively and individually, goes into moving and shaping material for such purposes, altering the physical landscape and our movement through it. Some of our most thrilling experiences come by way of changing our bodies' relation to that landscape: bungee jumping, skydiving, scuba diving, and riding a rollercoaster all provide radically new experiences for our bodies in the world.

As designers and builders of interactive systems for human use, we also play a central role in defining people's relationship to and experience of the physical world.[13,16] When we design things, we take mere ideas, things without form, and embody them in the world, whether simple sketches or cardboard mockups. They could be pixels on a screen or functioning digital devices. Regardless of the medium, to design and build things is to embody ideas that are then encountered and

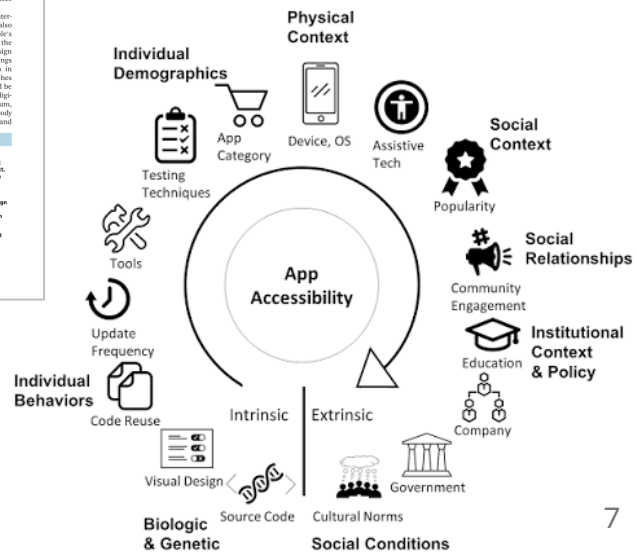
key insights

- Ability-based design is a new design approach for interactive systems that focuses on people's abilities in context, on what people can do, rather than on what they cannot do.
- Ability-based design scrutinizes the "ability assumptions" behind the design of interactive systems, shifting the responsibility of enabling access from users to the system.
- People's abilities may be affected not just by disabilities but by disabling situations; designing for abilities in context leads to more usable, accessible systems for all people.

62 COMMUNICATIONS OF THE ACM | JUNE 2018 | VOL. 61 | NO. 6

4. Pursue conceptual contributions

   Ability-Based Design (2010 – present)



5. Dive into other disciplines

   Epidemiology Model (2016 – 2022)

What about A.I.?



So, let’s dive in!



Remote Commander (2001 – 2002)











EdgeWrite

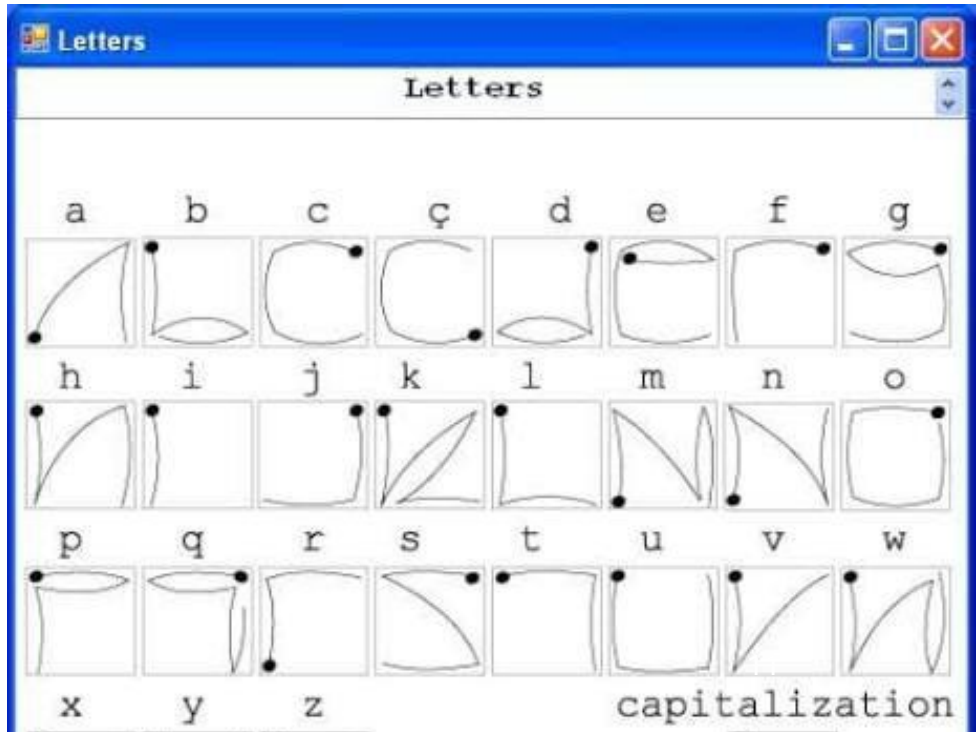
(2001 – 2006)



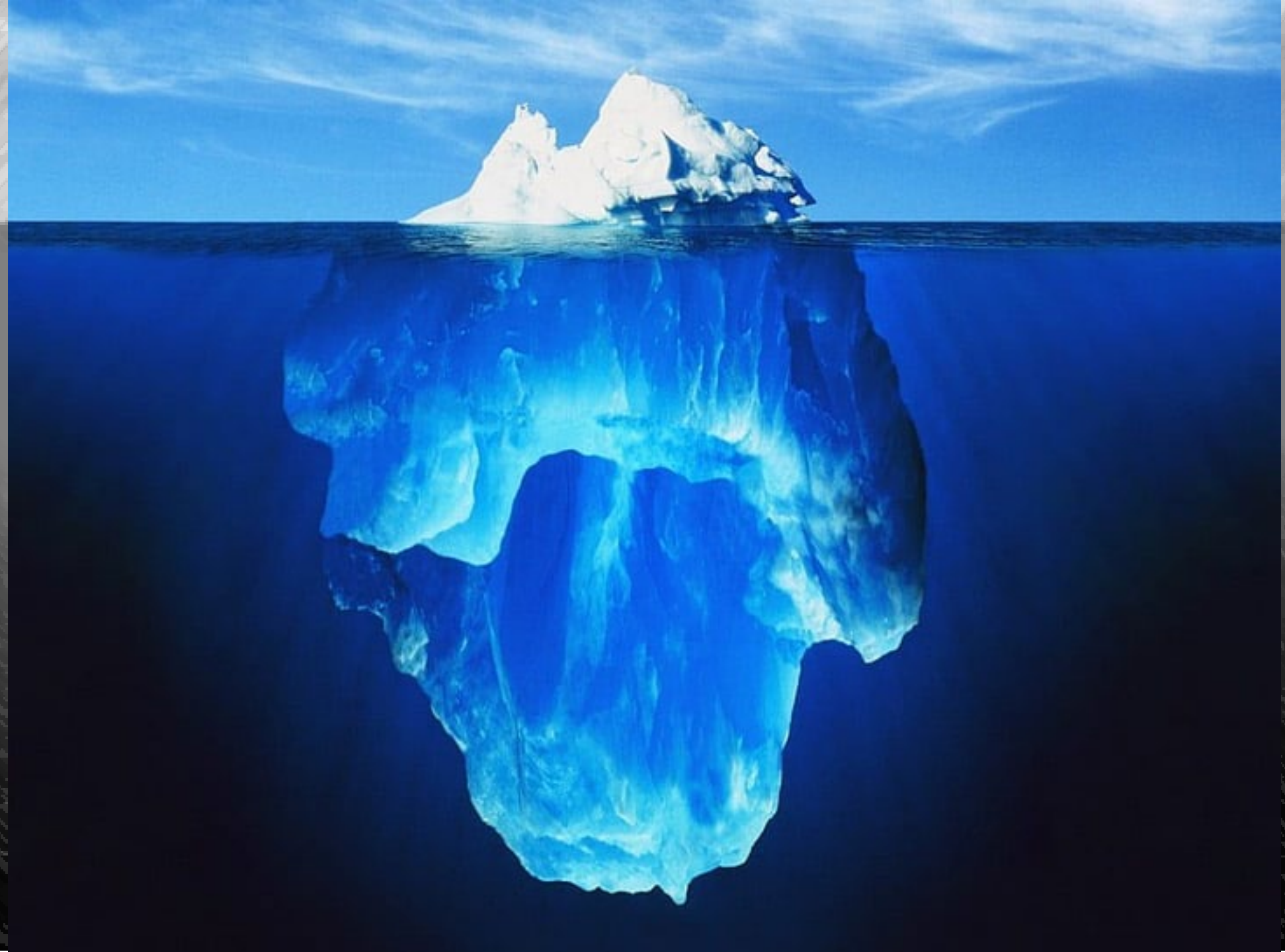






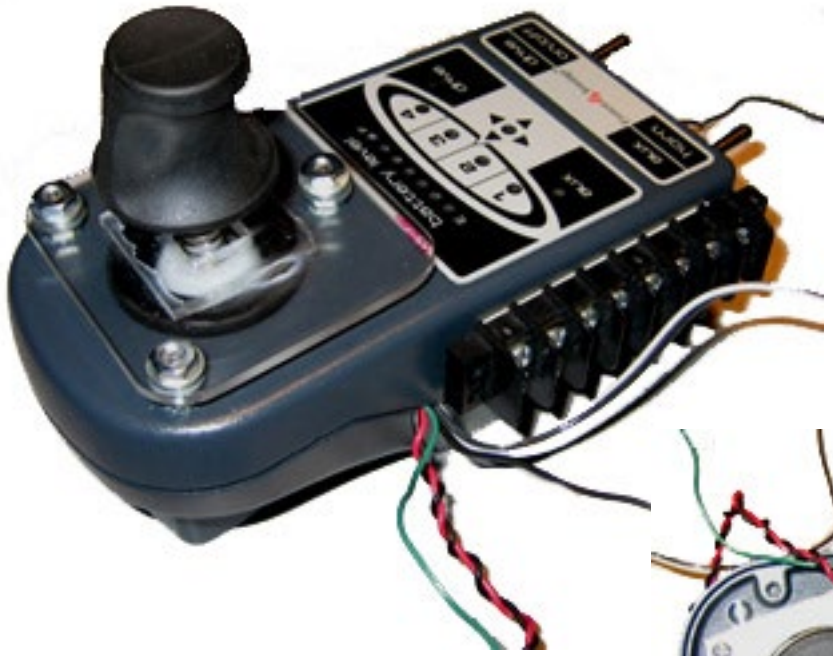


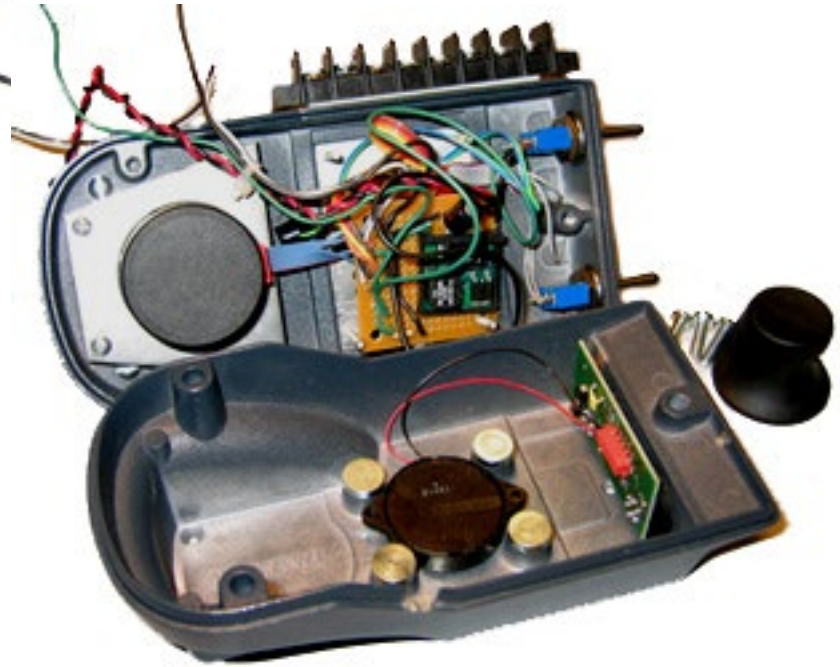




ASSETS ’04

(Best Paper Award)





Text Entry from Power Wheelchairs: EdgeWrite for Joysticks and Touchpads

Jacob O. Wobbrock,[1] Brad A. Myers,[1] Htet Htet Aung[1] and Edmund F. LoPresti[2]

[1]Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{ jrock, bam, hha }@cs.cmu.edu

[2]A.T. Sciences, Inc.
160 N. Craig Street, Suite 117
Pittsburgh, PA 15213 USA
(412) 687-1181
edlopresti@acm.org

ABSTRACT

Power wheelchair joysticks have been used to control a mouse cursor on desktop computers, but they offer no integrated text entry solution, confining users to point-and-click or point-and-dwell with on-screen keyboards. But on-screen keyboards reduce useful screen real-estate, exacerbate the need for frequent window management, and impose a second focus of attention. By contrast, we present two integrated gestural text entry methods designed for use from power wheelchairs: one for joysticks and the other for touchpads. Both techniques are adaptations of EdgeWrite, originally a stylus-based unistroke method designed for people with tremor. In a preliminary study of 7 power wheelchair users, we found that touchpad EdgeWrite was faster than joystick WiVik, and joystick EdgeWrite was only slightly slower after minimal practice. These findings reflect “walk up and use”-ability and warrant further investigation into extended use.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *input devices and strategies, user-centered design*

K.4.2 [**Computers and Society**]: Social Issues – *assistive technologies for persons with disabilities*

General Terms

Design, Experimentation, Human Factors

Keywords

Power wheelchair, computer access, joystick, touchpad, text entry, text input, unistrokes, gestures, EdgeWrite, Pebbles.

1. INTRODUCTION

People with motor impairments, such as those caused by Muscular Dystrophy, Cerebral Palsy, Parkinson’s disease, or spinal cord injuries, often cannot use a conventional mouse and keyboard. They may lack sufficient mobility to reach for these devices, sufficient motor control to switch accurately and efficiently between them, or sufficient endurance to use them for more than a few minutes. In addition, many people with motor impairments use wheelchairs. An estimated 1.4 million Americans depend on wheelchairs for mobility [14]. About 10% of these people use power wheelchairs, and about half of them require more than one assistive technology to participate in daily activities [4]. A computer access solution that works with an existing device, rather than adding to the mix of encumbering devices, would be desirable [11]. Such solutions have been termed “integrated control systems” [27].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ASSETS’04, October 18–20, 2004, Atlanta, Georgia, USA.
Copyright 2004 ACM 1-58113-911-X/04/0010...$5.00.



Figure 1. The Everest & Jennings 1706-5020 power wheelchair joystick we modified for EdgeWrite text entry. Note the plastic template around the stick, which provides a square boundary.

Commercial technology already exists for providing mouse cursor control from a power wheelchair joystick [28]. But mouse control is only part of a computer access solution. The ability to enter text is also a cornerstone of successful human-computer interaction. However, an integrated text entry method to accompany joystick mouse control is unavailable. Instead, text entry from power wheelchairs takes the form of point-and-click or point-and-dwell with on-screen keyboards. This exacerbates the need for window management due to decreased screen real-estate. It also imposes a secondary focus of attention, taking users’ eyes from their work. A text entry method for power wheelchair joysticks would give fuller access without requiring additional devices (Figure 1).

Though less common than joysticks, touchpads can also be used to control power wheelchairs [29] (Figure 2). Touchpads require less strength to operate than joysticks and no calibration. The further a finger moves from the center of the touchpad, the faster the wheelchair moves. While touchpads have been studied and

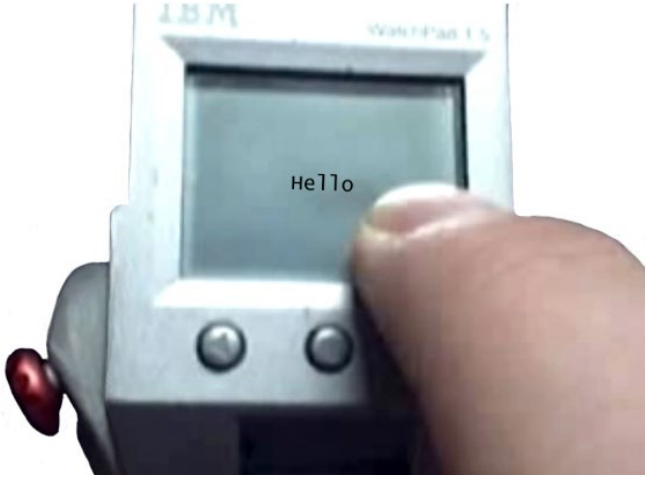

S

Others' versions

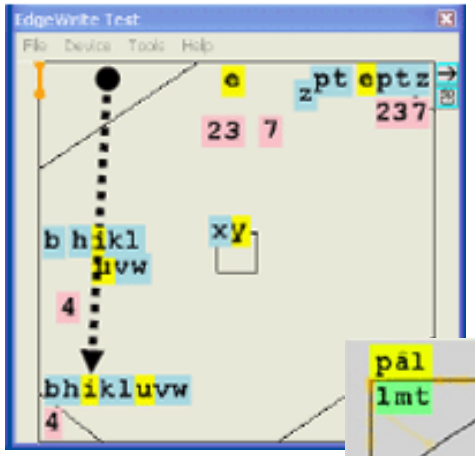
D-Shaped EdgeWrite: A Space-Saving Stabilized Text Entry Method for Round Face Smartwatches

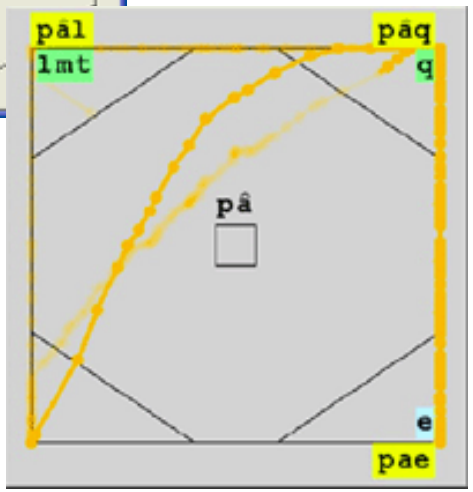
Conference paper | First Online: 01 June 2025

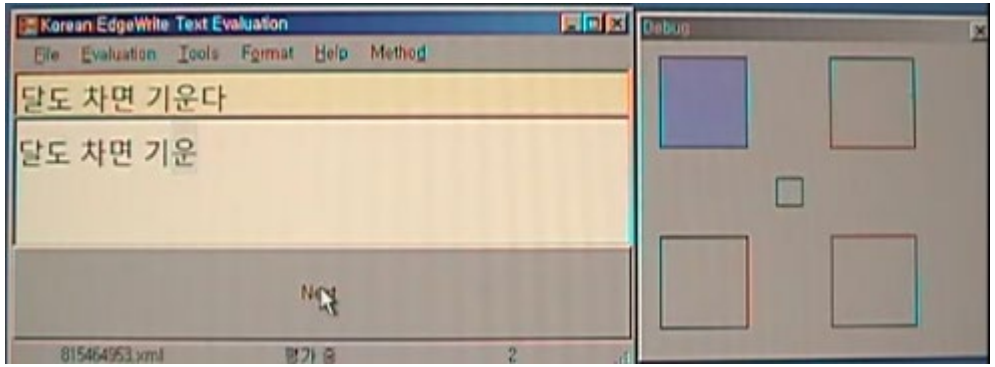
pp 3–15 | Cite this conference paper

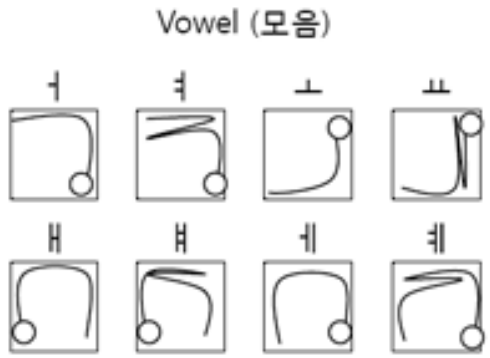


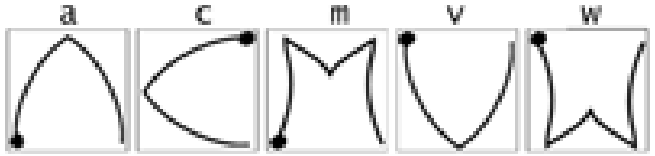


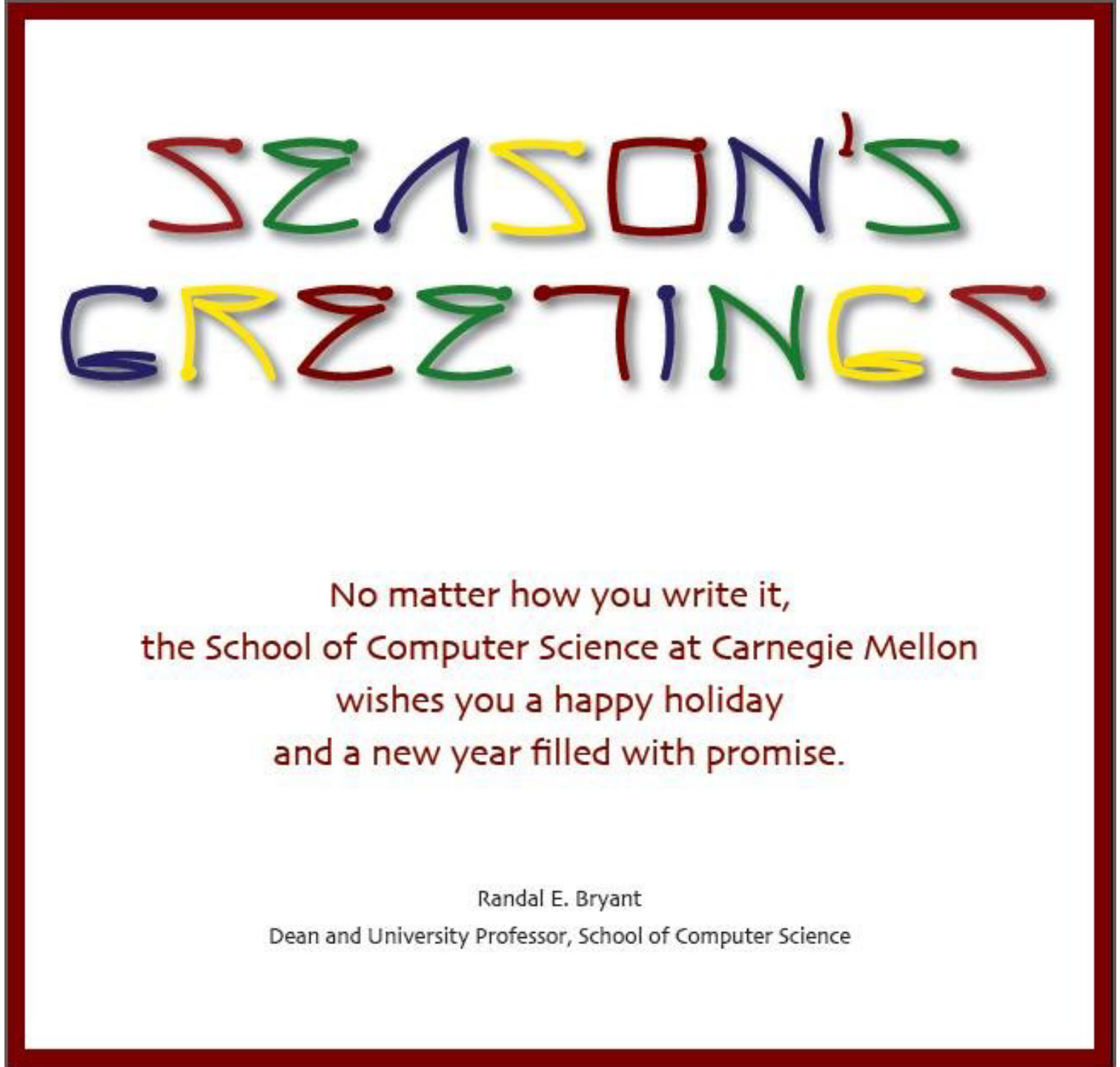











SEASON'S
GREETINGS
No matter how you write it,
the School of Computer Science at Carnegie Mellon
wishes you a happy holiday
and a new year filled with promise.
Randal E. Bryant
Dean and University Professor, School of Computer Science

Reflection: Prioritize depth

Don’t be afraid to take on the *whole iceberg*.

Stay with a topic long enough to fully explore it.

Novelty is not an end in itself.

Making discoveries and building solutions that address real accessibility problems for real people are the goals.



Flat glass? (2007)








A.I. Generated
Touch
Swipe
Explore

Participant-driven prototyping (2007-2008)

"Although our informants were experienced mobile device users, **most had not used many touch screens**. Informants reported using touch screens on devices such as microwave ovens, supermarket checkout kiosks, voting machines, ATMs, and occasionally, mobile devices, **but these were generally rare occasions**."

"While many of our informants were intrigued by the possibility of using an accessible touch screen, some were concerned about being **unable to find objects** on touch screens or **accidentally activating incorrect features**. Thus, it is important that touch screen interfaces are **easy to explore and minimize the need to search for on-screen items through trial-and-error**."

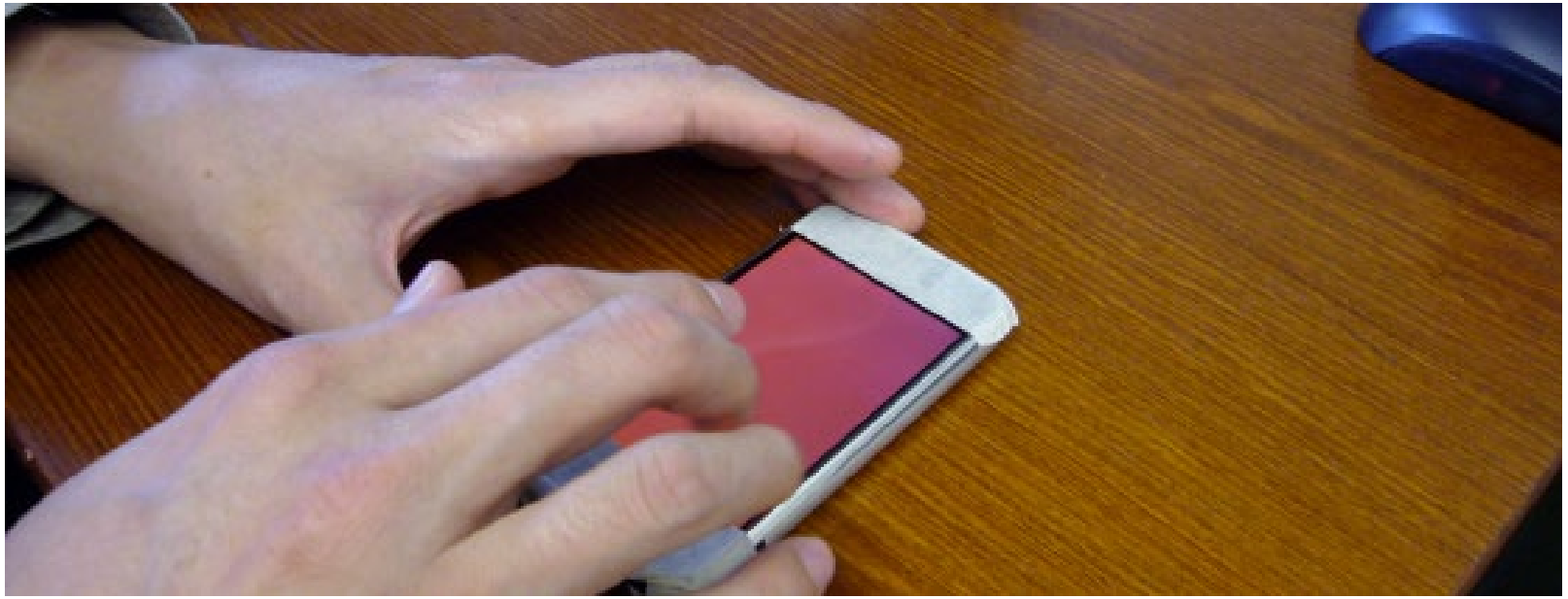
Shaun K. Kane et al., ASSETS '08

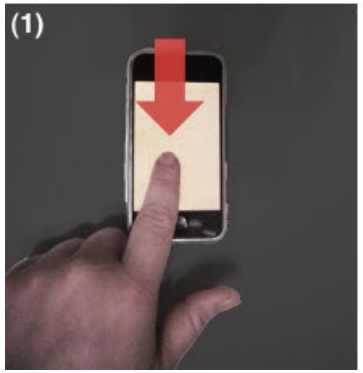


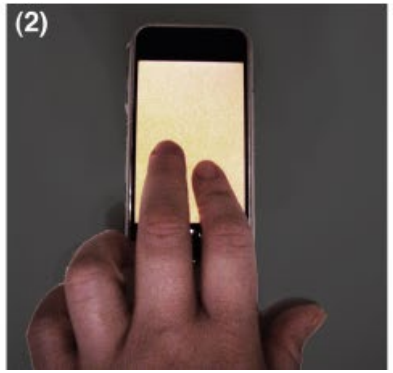
Participant-driven design principles

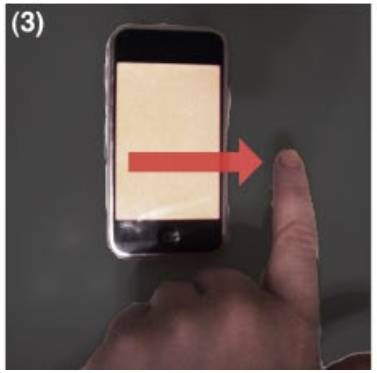
- Risk-free exploration
- Operate at finger resolution, not screen resolution
- Reduce demand for selection accuracy
- Enable quick browsing and navigation
- Employ intuitive gestural mappings
- Enable users to query their location and return home at any time from any place

Slide Rule (2008)

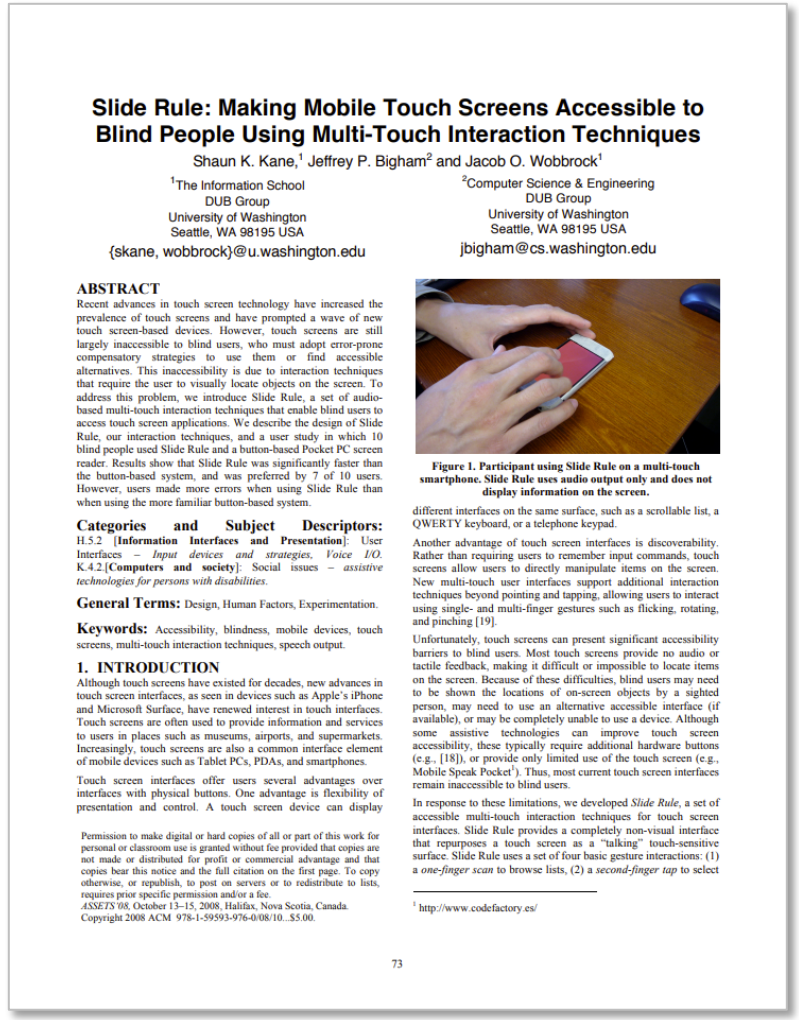












Slide Rule: Making Mobile Touch Screens Accessible to Blind People Using Multi-Touch Interaction Techniques

Shaun K. Kane,[1] Jeffrey P. Bigham[2] and Jacob O. Wobbrock[1]

[1]The Information School
DUB Group
University of Washington
Seattle, WA 98195 USA
{skane, wobbrock}@u.washington.edu

[2]Computer Science & Engineering
DUB Group
University of Washington
Seattle, WA 98195 USA
jbigham@cs.washington.edu

ABSTRACT

Recent advances in touch screen technology have increased the prevalence of touch screens and have prompted a wave of new touch screen-based devices. However, touch screens are still largely inaccessible to blind users, who must adopt error-prone compensatory strategies to use them or find accessible alternatives. This inaccessibility is due to interaction techniques that require the user to visually locate objects on the screen. To address this problem, we introduce Slide Rule, a set of audio-based multi-touch interaction techniques that enable blind users to access touch screen applications. We describe the design of Slide Rule, our interaction techniques, and a user study in which 10 blind people used Slide Rule and a button-based Pocket PC screen reader. Results show that Slide Rule was significantly faster than the button-based system, and was preferred by 7 of 10 users. However, users made more errors when using Slide Rule than when using the more familiar button-based system.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Input devices and strategies, Voice I/O.* K.4.2 [**Computers and society**]: Social issues – *assistive technologies for persons with disabilities.*

General Terms: Design, Human Factors, Experimentation.

Keywords: Accessibility, blindness, mobile devices, touch screens, multi-touch interaction techniques, speech output.

1. INTRODUCTION

Although touch screens have existed for decades, new advances in touch screen interfaces, as seen in devices such as Apple's iPhone and Microsoft Surface, have renewed interest in touch interfaces. Touch screens are often used to provide information and services to users in places such as museums, airports, and supermarkets. Increasingly, touch screens are also a common interface element of mobile devices such as Tablet PCs, PDAs, and smartphones.

Touch screen interfaces offer users several advantages over interfaces with physical buttons. One advantage is flexibility of presentation and control. A touch screen device can display

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ASSETS'08, October 13–15, 2008, Halifax, Nova Scotia, Canada.
Copyright 2008 ACM 978-1-59593-976-0/08/10...$5.00.

Figure 1. Participant using Slide Rule on a multi-touch smartphone. Slide Rule uses audio output only and does not display information on the screen.

different interfaces on the same surface, such as a scrollable list, a QWERTY keyboard, or a telephone keypad.

Another advantage of touch screen interfaces is discoverability. Rather than requiring users to remember input commands, touch screens allow users to directly manipulate items on the screen. New multi-touch user interfaces support additional interaction techniques beyond pointing and tapping, allowing users to interact using single- and multi-finger gestures such as flicking, rotating, and pinching [19].

Unfortunately, touch screens can present significant accessibility barriers to blind users. Most touch screens provide no audio or tactile feedback, making it difficult or impossible to locate items on the screen. Because of these difficulties, blind users may need to be shown the locations of on-screen objects by a sighted person, may need to use an alternative accessible interface (if available), or may be completely unable to use a device. Although some assistive technologies can improve touch screen accessibility, these typically require additional hardware buttons (e.g., [18]), or provide only limited use of the touch screen (e.g., Mobile Speak Pocket[1]). Thus, most current touch screen interfaces remain inaccessible to blind users.

In response to these limitations, we developed *Slide Rule*, a set of accessible multi-touch interaction techniques for touch screen interfaces. Slide Rule provides a completely non-visual interface that repurposes a touch screen as a "talking" touch-sensitive surface. Slide Rule uses a set of four basic gesture interactions: (1) a *one-finger scan* to browse lists, (2) a *second-finger tap* to select

[1] http://www.codefactory.es/



Slide Rule Success!

"We have been excited to see [VoiceOver] being so widely used amongst the visually impaired. We definitely read through the existing literature before starting [and] I can say we were certainly aware of this project [Slide Rule]. We were quite excited to see the video when it popped up."

– Apple Engineer, 10/16/2010



A small group of computer scientists first came up with an elegant solution to the problem of touch screen access. Doctoral student Shaun Kane, with help from fellow student Jeffrey Bigham and guidance from Kane's advisor Dr. Jacob Wobbrock, came up with what they called Slide Rule, a set of accessible gesture controls demonstrated on a touch screen mobile device (a modified iPhone, as it happened), in 2007 [35]. This technical solution made it possible for the blind to use a touch screen with a specific set of swipe gestures. Slide Rule technology was intriguing to the blind community to say the least. In March 2008, Kane and his team released developer tools for the iPhone (S. Kane, personal communication, March 11, 2014). The big question among the users was how to make the Slide Rule technology easier to obtain by average users. In 2009, the Apple iPhone 3GS was released. Integrated in the new device was VoiceOver, screen access software which provided speech feedback and gesture access to the iPhone. VoiceOver has been included in all subsequent releases of the iPhone. VoiceOver, the gestures for which bear some striking resemblances to Kane's work with Slide Rule, offered for the first time fully integrated native screen access software on a smartphone. With the interface, the blind finally had an accessible mobile phone to use without having to pay an additional cost for a speech program.

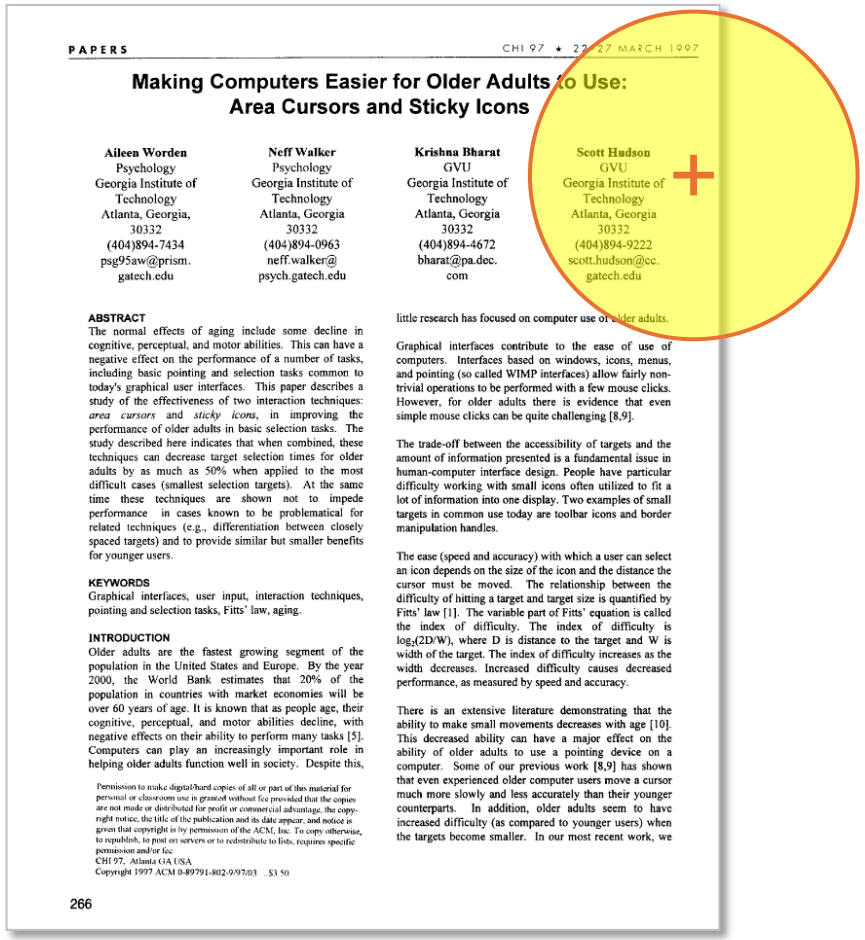


Reflection: Participant-driven prototyping

- Working extensively with participants is slow. It takes time.
- The pressure to produce the next new thing fast is increasing.
- A.I. makes it easier than ever to build rapid prototypes.
- But we must remain participant-driven in our prototyping.
- We must prototype *with*, and not just *for*, participants.
- Speed can be an enemy of true success.



The Persistent Problem of Pointing



PAPERS CHI 97 ★ 22-27 MARCH 1997

Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons

Aileen Worden Psychology Georgia Institute of Technology Atlanta, Georgia, 30332 (404)894-7434 psg95aw@prism.gatech.edu

Neff Walker Psychology Georgia Institute of Technology Atlanta, Georgia 30332 (404)894-0963 neff.walker@psych.gatech.edu

Krishna Bharat GVU Georgia Institute of Technology Atlanta, Georgia 30332 (404)894-4672 bharat@pa.dec.com

Scott Hudson GVU Georgia Institute of Technology Atlanta, Georgia 30332 (404)894-9222 scott.hudson@cc.gatech.edu

ABSTRACT

The normal effects of aging include some decline in cognitive, perceptual, and motor abilities. This can have a negative effect on the performance of a number of tasks, including basic pointing and selection tasks common to today's graphical user interfaces. This paper describes a study of the effectiveness of two interaction techniques: *area cursors* and *sticky icons*, in improving the performance of older adults in basic selection tasks. The study described here indicates that when combined, these techniques can decrease target selection times for older adults by as much as 50% when applied to the most difficult cases (smallest selection targets). At the same time these techniques are shown not to impede performance in cases known to be problematical for related techniques (e.g., differentiation between closely spaced targets) and to provide similar but smaller benefits for younger users.

KEYWORDS

Graphical interfaces, user input, interaction techniques, pointing and selection tasks, Fitts' law, aging.

INTRODUCTION

Older adults are the fastest growing segment of the population in the United States and Europe. By the year 2000, the World Bank estimates that 20% of the population in countries with market economies will be over 60 years of age. It is known that as people age, their cognitive, perceptual, and motor abilities decline, with negative effects on their ability to perform many tasks [5]. Computers can play an increasingly important role in helping older adults function well in society. Despite this, little research has focused on computer use of older adults.

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee
CHI 97, Atlanta GA USA
Copyright 1997 ACM 0-89791-802-9/97/03 ..$3.50

Graphical interfaces contribute to the ease of use of computers. Interfaces based on windows, icons, menus, and pointing (so called WIMP interfaces) allow fairly non-trivial operations to be performed with a few mouse clicks. However, for older adults there is evidence that even simple mouse clicks can be quite challenging [8,9].

The trade-off between the accessibility of targets and the amount of information presented is a fundamental issue in human-computer interface design. People have particular difficulty working with small icons often utilized to fit a lot of information into one display. Two examples of small targets in common use today are toolbar icons and border manipulation handles.

The ease (speed and accuracy) with which a user can select an icon depends on the size of the icon and the distance the cursor must be moved. The relationship between the difficulty of hitting a target and target size is quantified by Fitts' law [1]. The variable part of Fitts' equation is called the index of difficulty. The index of difficulty is $\log_2(2D/W)$, where D is distance to the target and W is width of the target. The index of difficulty increases as the width decreases. Increased difficulty causes decreased performance, as measured by speed and accuracy.

There is an extensive literature demonstrating that the ability to make small movements decreases with age [10]. This decreased ability can have a major effect on the ability of older adults to use a pointing device on a computer. Some of our previous work [8,9] has shown that even experienced older computer users move a cursor much more slowly and less accurately than their younger counterparts. In addition, older adults seem to have increased difficulty (as compared to younger users) when the targets become smaller. In our most recent work, we

Worden et al. (CHI 1997)



Enhanced Area Cursors: Reducing … for People with Motor …

Leah Findlater[1], Alex Jansen[1], Kristen Shi…
Peter Kamb[1], Joshua Rakita[1], Jacob …
[1]The Information School, [2]Computer Scien…
DUB Group, University of Was…
{ leahk…sen7, kshino, medixon, pkamb, rakita…

ABSTRACT

Computer users with motor impairmen… major challenges with conventional mouse pointing… challenges are mostly due to fine pointing corrections a… final stages of target acquisition. To reduce the need for correction-phase pointing and to lessen the effects of small target size on acquisition difficulty, we introduce four enhanced area cursors, two of which rely on magnification and two of which use goal crossing. In a study with motor-impaired and able-bodied users, we compared the new designs to the point and Bubble cursors, the latter of which had not been evaluated for users with motor impairments. Two enhanced area cursors, the *Visual-Motor-Magnifier* and *Click-and-Cross*, were the most successful new designs for users with motor impairments, reducing selection time for small targets by 19%, corrective submovements by 45%, and error rate by up to 82% compared to the point cursor. Although the Bubble cursor also improved performance, participants with motor impairments unanimously preferred the enhanced area cursors.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces—*Input devices and strategies*. K4.2. Computers and society: Social issues—*assistive technologies for persons with disabilities*.

General terms: Design, Human Factors

Keywords: Accessibility, area cursors, goal crossing, motor space, visual space, magnification, Bubble cursor.

INTRODUCTION

Computer users with motor impairments, such as low strength, intention tremor, poor coordination, and rapid fatigue, face major challenges with conventional mouse pointing. While pointing consists of both a ballistic phase and a corrective phase [24], the corrective phase can be particularly problematic for people with motor impairments where precision control is crucial [10,16,33]. As a result, small targets are especially difficult for users with motor impairments to acquire [6,32], and yet many such users still prefer to use commodity pointing devices for their

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
UIST'10, October 3–6, 2010, New York, New York, USA.
Copyright 2010 ACM 978-1-4503-0271-5/10/10…$10.00.

Figure 1. Two enhanced area cursors. Click-and-Cross: an area… cursor (top-left) transforms covered targets into crossing arcs (top-right). Visual-Motor-Magnifier: an area cursor (bottom-left) expands visual and motor space for point-and-click selection (bottom-right).

availability, low cost, and lack of stigma [6,11]. Unfortunately, many small targets exist in current desktop user interfaces: for example, links on a webpage may be only 8 pixels in height, while resizing borders on windows are much smaller (*e.g.*, 4 pixels in Windows XP, and 1 pixel in Microsoft Outlook Notes). Other problems involving small targets include slipping off targets, unintended clicks, and the effort and time required to physically perform a click [18,19,29,31].

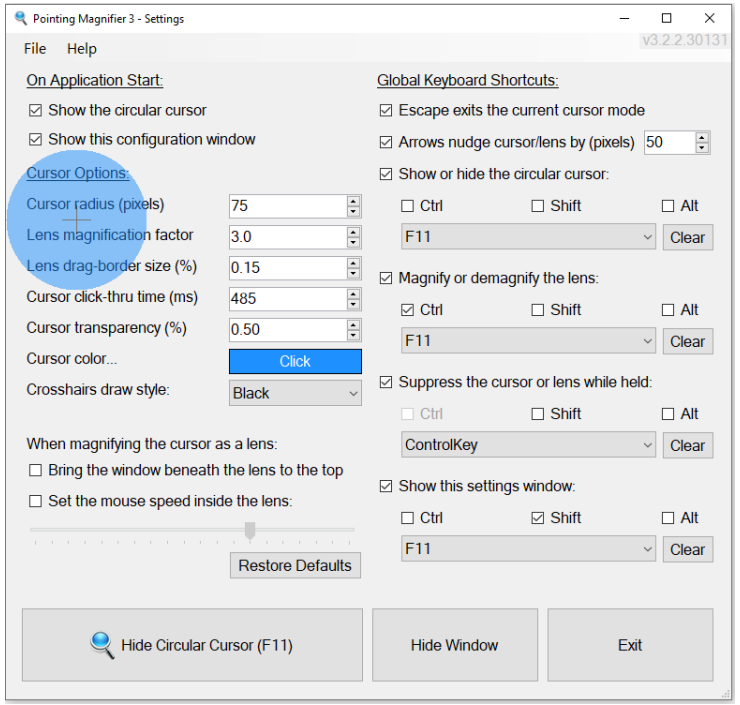
To reduce the need for fine pointing, we introduce *enhanced area cursors* (Figure 1). Unlike traditional area cursors [17,36], enhanced area cursors are specifically designed to work with small, dense targets, which is precisely where existing area cursors break down. Our techniques are based on the following design goals: (1) to reduce the need for corrective-phase pointing, (2) to lessen the effects of small targets on acquisition difficulty, and (3) to reduce the need for accurate, steady clicking, all without requiring specialized hardware. We present and evaluate two enhanced area cursors that use *goal crossing* instead of clicking (*Cross-and-Cross* and *Click-and-Cross*) and two cursors that provide *magnification* to ease selection (*Motor-Magnifier* and *Visual-Motor-Magnifier*).

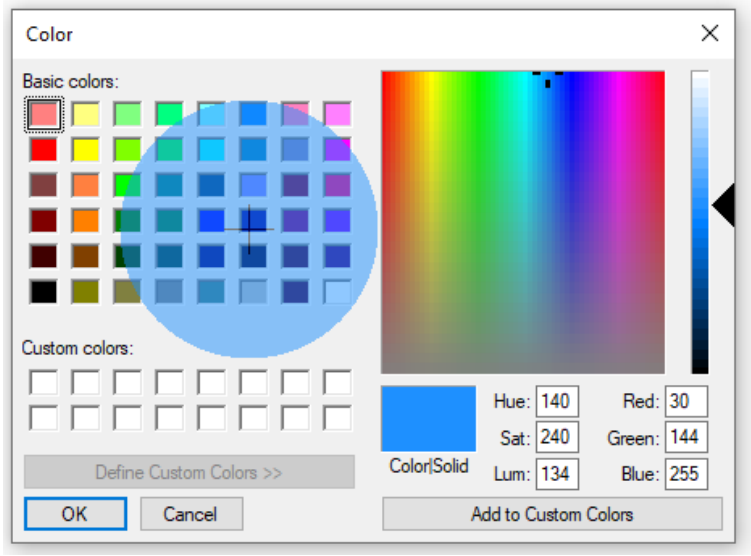
Although several techniques have been previously introduced to support target selection for users with motor impairments [15,30,34,35,36], many of these techniques degrade in small, dense target layouts—precisely those

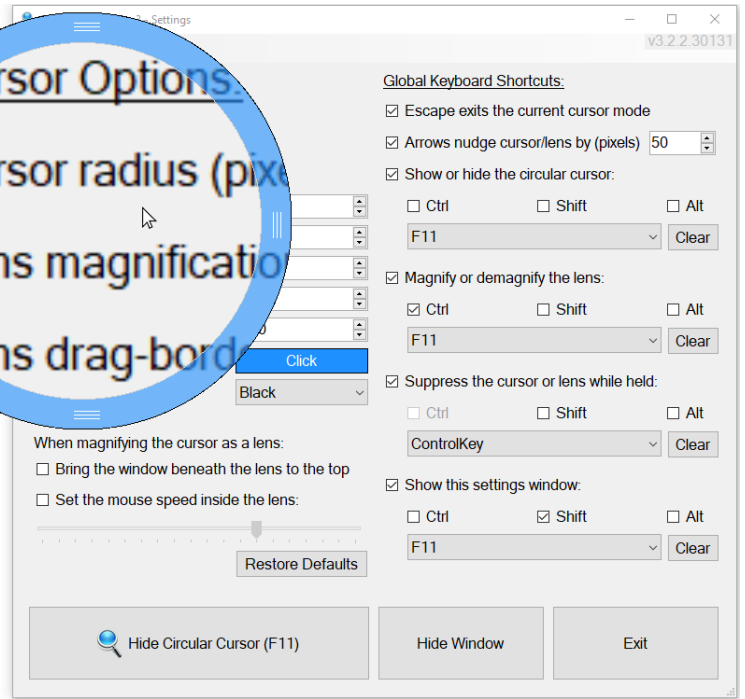
Findlater et al. (UIST 2010), Jansen et al. (CHI 2011)

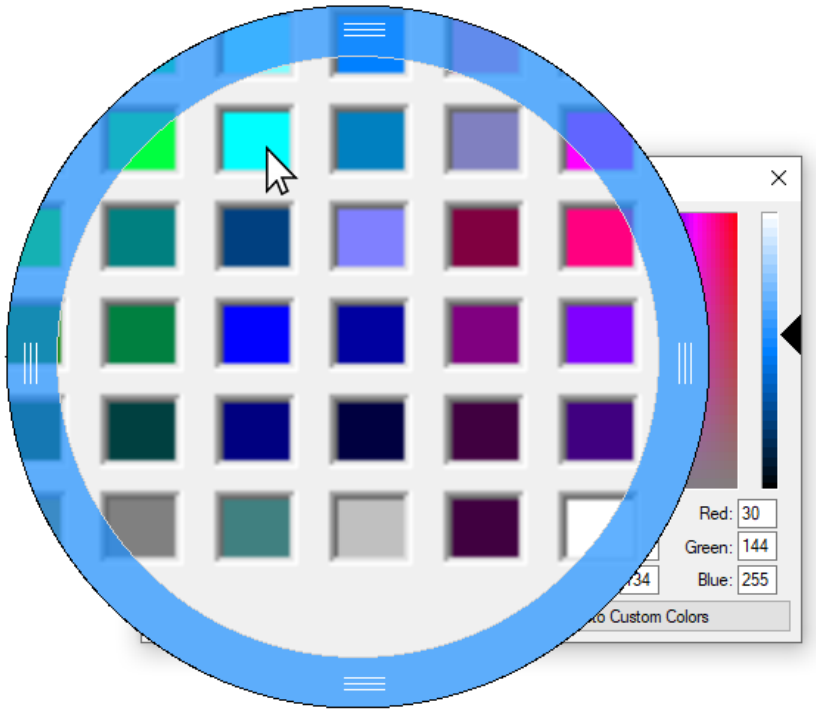


Pointing Magnifier



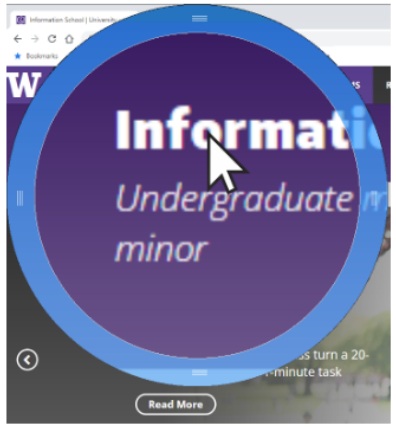






Pointing Magnifier 3

Jacob O. Wobbrock, Ph.D. [contact]
Leah Findlater, Ph.D.
Alexis Hiniker, Ph.D.
Alex Jansen
University of Washington



Click to enlarge

Download

Current Version 3.1.3.20587

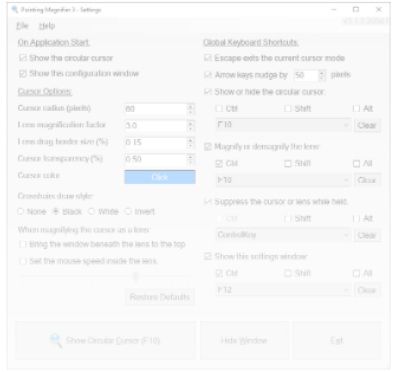
- Windows Installer: PM3.msi
- ZIP File: PM3.zip

You can choose to run the PM3.msi installer, or just download the PM3.zip file and unzip it. Inside, you will find the PM3.exe file, which can be run directly. If you choose to use the PM3.msi installer, note that you might see a Microsoft Defender warning when you install. This is expected until the installer builds reputation with Microsoft. You can proceed to install it anyway—we promise no harm will come!

The Pointing Magnifier 3 runs on the Microsoft .NET 4.8 Framework. After installing the application or unzipping the ZIP file, if the program does not run properly, you might need to Download the .NET 4.8 Runtime. (This should only be necessary on Windows computers that have not been updated via Windows Update.)

This software is distributed for non-commercial use under the New BSD License agreement. For commercial use, please contact Prof. Wobbrock at wobbrock@uw.edu.

About PM3 *(What's new!)* *(Known issues)*



Click to enlarge

The Pointing Magnifier is a cursor replacement designed to reduce the need for fine motor control. It is also useful in getting a close look at any region of the screen without losing the surrounding context, which has been useful to graphic designers and people with low vision. With the Pointing Magnifier, the usual mouse cursor is replaced by a semi-transparent circular "area cursor." When the user clicks, the area beneath this circular cursor becomes magnified, as if a lens. Once magnified, the typical mouse cursor arrow appears within the magnified lens, allowing for normal mouse operations, but now on much larger targets.

The Pointing Magnifier supports all button-based actions: clicking, double-clicking, right-clicking, dragging, and button dwelling, both on the cursor itself and within its magnified lens. It does not trigger mouse movement-based events—mouse over, mouse hover, mouse enter, or mouse leave—due to technical limitations.

In version 3 of the Pointing Magnifier, the lens itself can be dragged. It can also be nudged with the arrow keys. In addition, it contains support for multi-monitor displays.

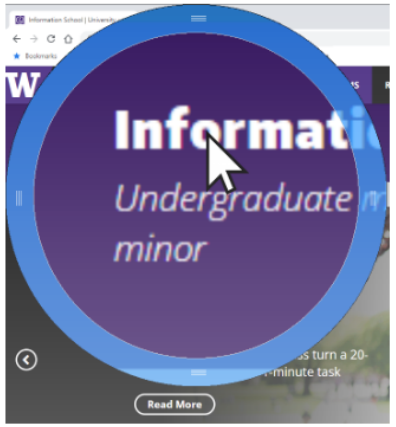
The original Pointing Magnifier was developed in 2010–2011 by Alex Jansen, undergrad designer Leah Findlater and supervised by Jacob O. Wobbrock. More than a decade later, numerous features requested in design

~175k downloads



Pointing Magnifier 3

Jacob O. Wobbrock, Ph.D. [contact]
Leah Findlater, Ph.D.
Alexis Hiniker, Ph.D.
Alex Jansen
University of Washington



Click to enlarge

Download

Current Version 3.1.3.20587

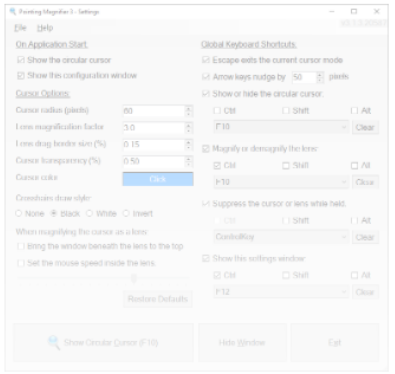
- Windows Installer: PM3.msi
- ZIP File: PM3.zip

You can choose to run the PM3.msi installer, or just download the PM3.zip file and unzip it. Inside, you will find the PM3.exe file, which can be run directly. If you choose to use the PM3.msi installer, note that you might see a Microsoft Defender warning when you install. This is expected until the installer builds reputation with Microsoft. You can proceed to install it anyway—we promise no harm will come!

The Pointing Magnifier 3 runs on the Microsoft .NET 4.8 Framework. After installing the application or unzipping the ZIP file, if the program does not run properly, you might need to Download the .NET 4.8 Runtime. (This should only be necessary on Windows computers that have not been updated via Windows Update.)

This software is distributed for non-commercial use under the New BSD License agreement. For commercial use, please contact Prof. Wobbrock at wobbrock@uw.edu.

About PM3 *(What's new!)* *(Known issues)*



Click to enlarge

The Pointing Magnifier is a cursor replacement designed to reduce the need for fine motor control. It is also useful in getting a close look at any region of the screen without losing the surrounding context, which has been useful to graphic designers and people with low vision. With the Pointing Magnifier, the usual mouse cursor is replaced by a semi-transparent circular "area cursor." When the user clicks, the area beneath this circular cursor becomes magnified, as if a lens. Once magnified, the typical mouse cursor arrow appears within the magnified lens, allowing for normal mouse operations, but now on much larger targets.

The Pointing Magnifier supports all button-based actions: clicking, double-clicking, right-clicking, dragging, and button dwelling, both on the cursor itself and within its magnified lens. It does not trigger mouse movement-based events—mouse over, mouse hover, mouse enter, or mouse leave—due to technical limitations.

In version 3 of the Pointing Magnifier, the lens itself can be dragged. It can also be nudged with the arrow keys. In addition, it contains support for multi-monitor displays.

The original Pointing Magnifier was developed in 2010-2011 by Alex Jansen, with lead designer Leah Findlater and supervised by Jacob O. Wobbrock. More than a decade later, numerous feature requests and design

This is a tremendous program. Incredibly easy to install and use. It is absolutely the kind of thing people with eyesight issues could benefit from**. I think the real benefactors will be people with hand tremors. It's exactly what we need** and I like that it is so easy to customize.

– Ari Wilkenfeld

I have been looking for a good and overall user-friendly magnification program for my Windows 10 machine. I happened to stumble upon yours and downloaded it and tried it and I hate to say, but **I'm almost in love with it.** The program is simple to use and **basically everything I need/want** in a magnifier.

– John Bresn

Read your article in *Mechanical Engineering*, June 2011. Thank you!! **I really need something like this. Getting shakey, & losing visual acuity.** Goes with old age!

– Joseph J. Lafranca, Jr.

I have the onset of Parkinson's and find it increasingly difficult to position the cursor. **I find your program helps with positioning.** Thanks again. Kind regards,

– Tony Phillips

Reflection: Deploy and Support in the Real World

- If we want to have impact, we need to get “real things” into the hands of “real users” as often as possible.
- But we also incur ethical obligations when we do.
- “Publish and move on” does not support the people who come to rely on our inventions.
- But we researchers aren’t set up to provide customer support and ongoing maintenance.
- We need to find better models for maintaining our inventions when people come to rely on them.
- But we should not let this obligation deter us from releasing useful things into the world.



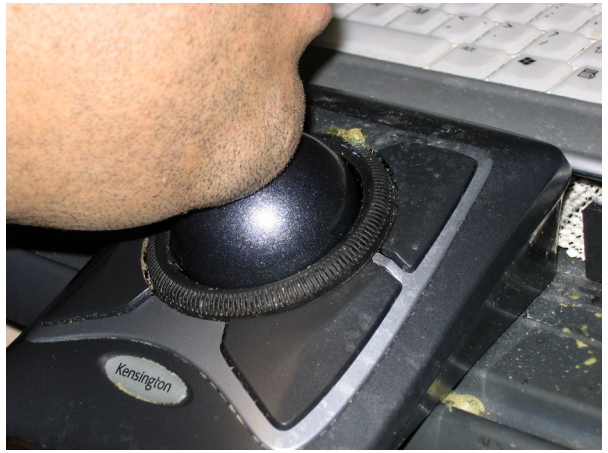

A.I. Generated

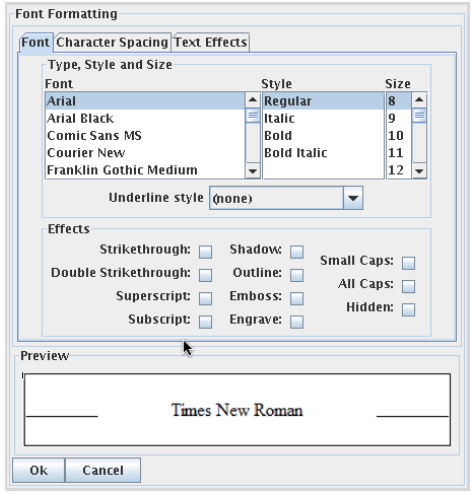

A.I. Generated

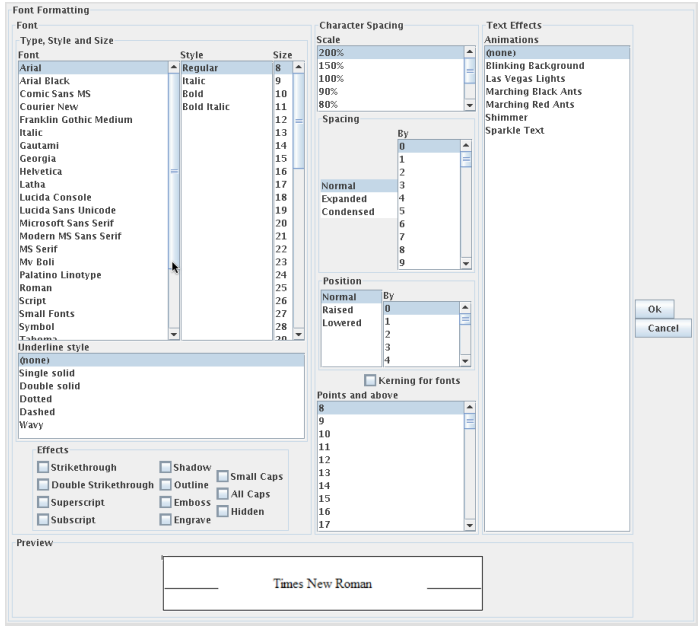
SUPPLE

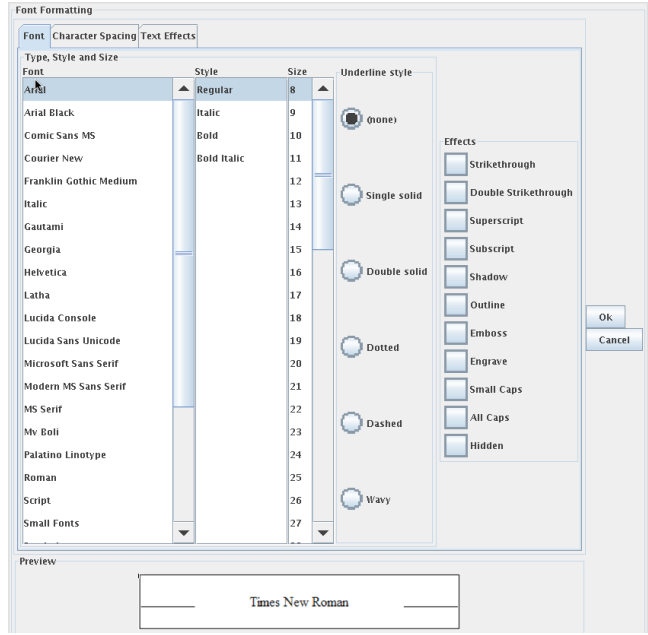
(Gajos et al. 2004-2010)











Focus on what people *can do*, not what they *can't do*.

Scrutinize interfaces for their *ability assumptions*…

Ensure other abilities can be used successfully with those interfaces.

Shift from "restoring function" to "observing, understanding, and adapting to what users *can do*."



Sense and respond to contexts, situations, and environments.

Design for ability, not dis-ability.

Ability-Based Design (2011, 2018)



Ability-Based Design: Concept, Principles and Examples

JACOB O. WOBBROCK and SHAUN K. KANE, University of Washington
KRZYSZTOF Z. GAJOS, Harvard School of Engineering and Applied Sciences
SUSUMU HARADA and JON FROEHLICH, University of Washington

Current approaches to accessible computing share a common goal of making technology accessible t
with disabilities. Perhaps because of this goal, they may also share a tendency to centralize di
rather than ability. We present a refinement to these approaches called *ability-based design* that cor
focusing on ability throughout the design process in an effort to create systems that leverage the full r
human potential. Just as user-centered design shifted the focus of interactive system design from sys
users, ability-based design attempts to shift the focus of accessible design from disability to ability. A
prior approaches to accessible computing may consider users' abilities to some extent, ability-based
makes ability its central focus. We offer seven ability-based design principles and describe the proje
inspired their formulation. We also present a research agenda for ability-based design.

Categories and Subject Descriptors: K.4.2 [**Computers and Society**]: Social Issues—*Assistive techr*
for persons with disabilities

General Terms: Design, Human Factors, Theory

Additional Key Words and Phrases: Ability-based design, inclusive design, universal design, unive
ability, design for all, user interfaces for all, computer access, assistive technology, adaptive user int

ACM Reference Format:
Wobbrock, J. O., Kane, S. K., Gajos, K. Z., Harada, S., and Froehlich, J. 2011. Ability-based design: C
principles and examples. ACM Trans. Access. Comput. 3, 3, Article 9 (April 2011), 27 pages.
DOI = 10.1145/1952383.1952384 http://doi.acm.org/10.1145/1952383.1952384

1. INTRODUCTION

Despite more than twenty years of research in accessible computing, user inte
still pose access challenges for many people with disabilities. Ironically, a cons
centralization of *disability* may be partly to blame. By its very name, a *dis*-abi
not so much a thing as the lack of a thing, namely the lack of ability. Takin
perspective, a person with a disability is *un*-able to perform certain everyday
and requires assistance to compensate for his or her limitation. However, peo
not have disabilities any more than they have dis-money or dis-height. Abilitie

This work was supported in part by the National Science Foundation under grants IIS-0952786 a
0811063. Any opinions, findings, conclusions or recommendations expressed in this work are thos
authors and do not necessarily reflect those of the National Science Foundation.
S. Harada is currently employed at IBM Research-Tokyo.
Authors' addresses: J. O. Wobbrock, Information School, University of Washington, Seattle, WA 9819
email: wobbrock@uw.edu; S. K. Kane; email: skane@uw.edu; K. Z. Gajos, Harvard School of Engi
and Applied Sciences, Cambridge, MA 02138; email: kgajos@eecs.harvard.edu; S. Harada, 1623-
motsuruma, Yamato-shi, Kanagawa-ken, 242-8502, Japan; email: harada@gmail.com; J. Froehlich
jfroehli@cs.washington.edu.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is
without fee provided that copies are not made or distributed for profit or commercial advantage a
copies show this notice on the first page or initial screen of a display along with the full citation. Cop
for components of this work owned by others than ACM must be honored. Abstracting with credit is
ted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any compo
this work in other works requires prior specific permission and/or a fee. Permissions may be request
the Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +
869-0481, or permissions@acm.org.
© 2011 ACM 1936-7228/2011/04-ART9 $10.00
DOI 10.1145/1952383.1952384 http://doi.acm.org/10.1145/1952383.1952384



DOI:10.1145/3148051

By focusing on users' abilities rather than disabilities, designers can create interactive systems better matched to those abilities.

BY JACOB O. WOBBROCK, KRZYSZTOF Z. GAJOS, SHAUN K. KANE, AND GREGG C. VANDERHEIDEN

Ability-Based Design

RECALL THE LAST time you took a trip out of town. Perhaps you were traveling to a conference far from home. Remember the many forms of transportation you endured: cars, buses, airplanes, and trains. Not only were you responsible for moving yourself over a great distance, you had to move your things as well, including books and baggage. Remember the cramped spaces, sharp elbows, body aches, and exhaustion. Feel again your desire to simply be at your destination with your possessions intact . . .

Such journeys remind us of our physical embodiment in the physical world, that much of our lived experience is fundamentally physical, and that we must contend with the world on physical terms. As computing professionals, we might be tempted to forget this, as our keystrokes summon data instantly from across the globe. But as humans, we still interact with that data through physical devices and displays using our physical senses and bodies. We and the world interact physically.

Civilization's story of technological progress is in no small part the story of an increasingly built physical environment, from the pyramids to roads to skyscrapers to sanitation systems. Much of our energy, collectively and individually, goes into moving and shaping material for such purposes, altering the physical landscape and our movement through it. Some of our most thrilling experiences come by way of changing our bodies' relation to that landscape: bungee jumping, skydiving, scuba diving, and riding a rollercoaster all provide radically new experiences for our bodies in the world.

As designers and builders of interactive systems for human use, we also play a central role in defining people's relationship to and experience of the physical world.[2,13,30] When we design things, we take mere ideas, things without form, and embody them in the world, whether simple sketches or cardboard mockups. They could be pixels on a screen or functioning digital devices. Regardless of the medium, to design and build things is to embody ideas that are then encountered and

» key insights

- **Ability-based design is a new design approach for interactive systems that focuses on people's abilities in context, on what people *can* do, rather than on what they *cannot* do.**
- **Ability-based design scrutinizes the "ability assumptions" behind the design of interactive systems, shifting the responsibility of enabling access from users to the system.**
- **People's abilities may be affected not just by disabilities but by disabling situations; designing for abilities in context leads to more usable, accessible systems for all people.**

Microsoft

Google.org

2017 SIGCHI Social Impact Award



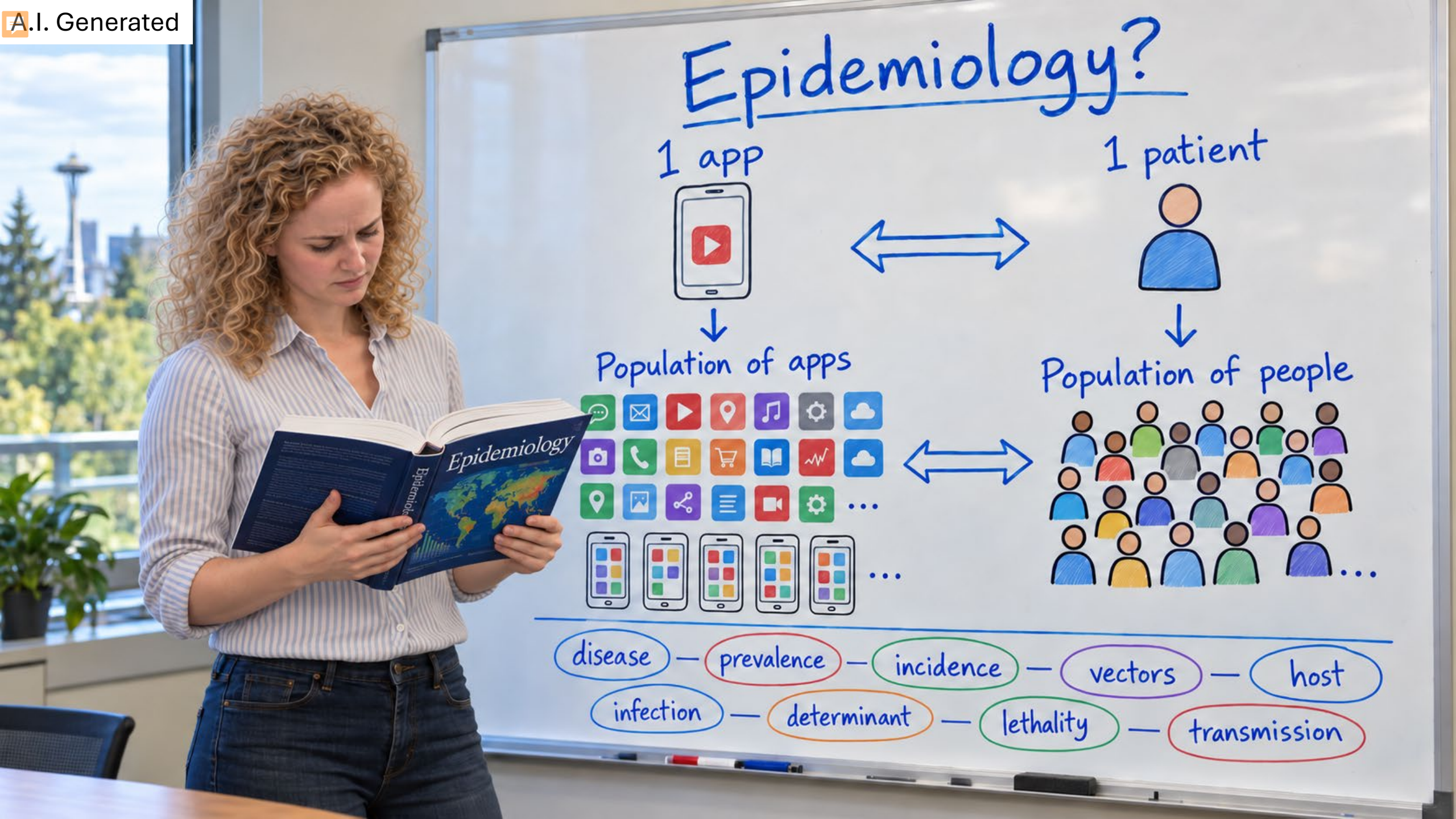
Reflection: Pursue Conceptual Contributions

- Conceptual contributions are crucial to how we think and work.
- “Zoom out” to consider what we’re building, and why and how we’re doing so.
- Think beyond artifacts to lasting concepts and enduring methods.
- Methods, theories, visions…. All can have lasting impact.
- They can even be cited in 20, 50, or 100 years!




A.I. Generated
Mobile accessibility
1 app
10,000 apps
Google Play Store
Labels
Touch targets
Contrast

A.I. Generated




natural history of a disease
infectious agent
life expectancy
prevalence
reservoir
disease
health
host
population
incidence
contagiousness
case
epidemiology
lethality
determinant
factor
diagnosis
mortality rate
accessibility
high-risk group
health indicator
usual source of care
public health
transmission
herd immunity
therapy
census
outbreak
treatment
prevention
common source outbreak
detection bias
universal precautions

10,000 apps!

Epidemiology as a Framework for Large-Scale Mobile Application Accessibility Assessment

Anne Spencer Ross[1], Xiaoyi Zhang[1], James Fogarty[1], Jacob O. Wobbrock[2]
[1]Computer Science & Engineering, [2]The Information School
DUB Group, University of Washington Seattle, WA 98195 USA
{ansross, xiaoyiz, jfogarty}@cs.washington.edu, wobbrock@uw.edu

ABSTRACT

Mobile accessibility is often a property considered at the level of a single mobile application (app), but rarely on a larger scale of the entire app "ecosystem," such as all apps in an app store, their companies, developers, and user influences. We present a novel conceptual framework for the accessibility of mobile apps inspired by epidemiology. It considers apps within their ecosystems, over time, and at a population level. Under this metaphor, "inaccessibility" is a set of *diseases* that can be viewed through an epidemiological lens. Accordingly, our framework puts forth notions like *risk* and *protective factors*, *prevalence*, and *health indicators* found within a *population* of apps. This new framing offers terminology, motivation, and techniques to reframe how we approach and measure app accessibility. It establishes how app accessibility can benefit from multi-factor, longitudinal, and population-based analyses. Our epidemiology-inspired conceptual framework is the main contribution of this work, intended to provoke thought and inspire new work enhancing app accessibility at a systemic level. In a preliminary exercising of our framework, we perform an analysis of the *prevalence* of common *determinants* or accessibility barriers. We assess the *health* of a stratified sample of 100 popular Android apps using Google's Accessibility Scanner. We find that 100% of apps have at least one of nine accessibility errors and examine which errors are most common. A preliminary analysis of the frequency of co-occurrences of multiple errors in a single app is also presented. We find 72% of apps have five or six errors, suggesting an interaction among different errors or an underlying influence.

CCS Concepts

• **Human-Centered Computing→Accessibility→Accessibility Theory, Concepts and Paradigms**

Keywords

Mobile computing; mobile accessibility; app accessibility; accessibility assessment; conceptual framework; epidemiology.

1. INTRODUCTION

Mobile applications (apps) play increasingly important roles in many aspects of life, including personal finances, communication, community engagement, and transportation. Supporting access to these impactful technologies for all people is important. Yet, for the approximately 15% of the world population with disabilities [35], many of these apps are inaccessible. Understanding the extent and causes of app inaccessibility is imperative for creating successful solutions. To date, most approaches to understanding and improving app accessibility have focused on specific issues within individual apps, such as developer guidelines [1,4], app scanners [5,19], and compiler warnings [2]. But apps do not exist in isolation. They might come from the same developers or studios, utilize the same libraries or code repositories, mimic the same popular design patterns, or simply be subsequent versions of prior releases. These factors can perpetuate accessibility problems, but the factors are rarely, if ever, analyzed at a "multi-factor" or "population level" over myriad apps. Expanding how we approach app accessibility to encompass the rich ecosystem in which apps exist can drive innovations in how we work to enhance accessibility. Fortunately, other areas of study provide a model for multi-factor population-level analyses. One such area is *epidemiology*, which examines health on a population level from a systems perspective. Whereas a physician treats a single patient, an epidemiologist considers the health of a whole population of patients. Analogously, a designer or developer

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ASSETS '17, October 29-November 1, 2017, Baltimore, MD, USA
© 2017 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-4926-0/17/10...$15.00
https://doi.org/10.1145/3132525.3132547



Figure 1. As a systems science, epidemiology can serve as a metaphor that changes the way we think and work with mobile app accessibility. Here, the concept of a multi-factor ecosystem from epidemiology has been applied to mobile app accessibility. An app's accessibility is a product of many factors ranging from individual and intrinsic to population-level and extrinsic. These factors include: source code and design, behaviors, demographics, physical context, social context and relationships, institutional context and policies, and cultural norms. Accessibility is affected by factors at all levels. Figure inspired by [34].

Examining Image-Based Button Labeling for Accessibility in Android Apps through Large-Scale Analysis

Anne Spencer Ross[1], Xiaoyi Zhang[1], James Fogarty[1], Jacob O. Wobbrock[2]
[1]Paul G. Allen School of Computer Science & Engineering, [2]The Information School
DUB Group, University of Washington, Seattle, WA, USA
{ansross, xiaoyiz, jfogarty}@cs.washington.edu, wobbrock@uw.edu

ABSTRACT

We conduct the first large-scale analysis of the accessibility of mobile apps, examining what unique insights this can provide into the state of mobile app accessibility. We analyzed 5,753 free Android apps for label-based accessibility barriers in three classes of image-based buttons: Clickable Images, Image Buttons, and Floating Action Buttons. An epidemiology-inspired framework was used to structure the investigation. The *population* of free Android apps was assessed for label-based inaccessible button *diseases*. Three *determinants* of the *disease* were considered: missing labels, duplicate labels, and uninformative labels. The *prevalence*, or frequency of occurrences of barriers, was examined in apps and in classes of image-based buttons. In the app analysis, 35.9% of analyzed apps had 90% or more of their assessed image-based buttons labeled, 45.9% had less than 10% of assessed image-based buttons labeled, and the remaining apps were relatively uniformly distributed along the proportion of elements that were labeled. In the class analysis, 92.0% of Floating Action Buttons were found to have missing labels, compared to 54.7% of Image Buttons and 86.3% of Clickable Images. We discuss how these accessibility barriers are addressed in existing *treatments*, including accessibility development guidelines.

Author Keywords

Mobile app; large-scale analysis; accessibility; image-based buttons

ACM Classification Keywords

• **Human-centered computing–Empirical studies in accessibility** • Human-centered computing–Heuristic evaluations

INTRODUCTION

Mobile applications (apps) are becoming increasingly important in daily life, providing information and services in a range of settings that include banking, communication, education, entertainment and travel. It is important that these powerful capabilities are available to all people, regardless of their abilities or use of assistive technologies. However, prior work has shown there still exist important accessibility barriers within apps [20,21,37,40,42,43].

Awareness of the need to create more accessible apps is increasing. Google and Apple are the primary organizations that facilitate mobile technology and the app marketplace, through the Android and iOS platforms. Both have released developer and design guidelines for accessibility [16,30], provide accessibility services as part of their platforms [2,15], have app development libraries that include built-in compatibility with assistive technologies, and have released accessibility testing scanners [17,28] and suites [10]. Some companies creating popular apps have also made statements and taken actions to create more accessible apps [19,45]. Such approaches have included creating internal guidelines [45], having specialized accessibility teams [19], actively prioritizing accessibility [19,45], and working with people with disabilities during app development and testing [45].

Despite these accessibility-focused efforts, studies of relatively small groups of apps have found they still include significant accessibility barriers [20,21,37,42,43]. This suggests a continuing need for accessibility improvements, however, the field lacks a detailed understanding of the state of mobile app accessibility at a large-scale, "population" level.

Many design patterns in mobile apps are image and icon focused and use image-based buttons for main functionalities. One key component of accessibility for screen reader users is labeling image-based buttons. This need is parallel to the need for alt-text for images on the web. However, there is no large-scale understanding of the prevalence of unlabeled image-based buttons, how effective tools are at promoting labeling, nor the potential causes of failure to label. Ross *et al.* [42] present an epidemiology-based framework that suggests large-scale analyses can help answer some of these questions. The framework emphasizes that apps do not exist in isolation. It suggests that, in addition to testing individual apps, additional benefits and insights can be gained by exploring app accessibility at the population-level, situated within the richer ecosystem of influential factors. Such analyses can give unique insights into the state of app accessibility and opportunities for improvements. An epidemiologically-inspired analysis can also establish a baseline against which to measure the evolution of app accessibility over time.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
ASSETS '18, October 22–24, 2018, Galway, Ireland
© 2018 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-5650-3/18/10…$15.00
https://doi.org/10.1145/3234695.3236364

An Epidemiology-inspired Large-scale Analysis of Android App Accessibility

ANNE SPENCER ROSS, XIAOYI ZHANG, and JAMES FOGARTY, Paul G. Allen School of Computer Science & Engineering, DUB Group, University of Washington, Seattle, WA, USA
JACOB O. WOBBROCK, The Information School, DUB Group, University of Washington, Seattle, WA, USA

Accessibility barriers in mobile applications (apps) can make it challenging for people who have impairments or use assistive technology to use those apps. Ross et al.'s epidemiology-inspired framework emphasizes that a wide variety of factors may influence an app's accessibility and presents large-scale analysis as a powerful tool for understanding the prevalence of accessibility barriers (i.e., *inaccessibility diseases*). Drawing on this framework, we performed a large-scale analysis of free Android apps, exploring the frequency of accessibility barriers and factors that may have contributed to barrier prevalence. We tested a population of 9,999 apps for seven accessibility barriers: few TalkBack-focusable elements, missing labels, duplicate labels, uninformative labels, editable TextViews with *contentDescription*, fully overlapping clickable elements, and undersized elements. We began by measuring the prevalence of each accessibility barrier across all relevant element classes and apps. Missing labels and undersized elements were the most prevalent barriers. As a measure of the spread of barriers across apps, we assessed the five most reused classes of elements for missing labels and undersized elements. The Image Button class was among the most barrier-prone of the high reuse element classes; 53% of Image Button elements were missing labels and 40% were undersized. We also investigated factors that may have contributed to the high barrier prevalence in certain classes of elements, selecting examples based on prior knowledge, our analyses, and metrics of reuse and barrier-proneness. These case studies explore: (1) how the *few TalkBack-focusable elements* accessibility barrier relates to app category (e.g., Education, Entertainment) and the tools used to implement an app, (2) the prevalence of label-based barriers in image-based buttons, (3) design patterns that affect the labeling and size of Radio Buttons and Checkboxes, and (4) accessibility implications of the sizing of third-party plug-in elements. Our work characterizes the current state of Android accessibility, suggests improvements to the app ecosystem, and demonstrates analysis techniques that can be applied in further app accessibility assessments.

CCS Concepts: • **Human-centered computing → Empirical studies in accessibility**; **Accessibility design and evaluation methods**;

Additional Key Words and Phrases: Mobile applications, large-scale analyses, accessibility

X. Zhang is now at Apple, Inc.
This work was funded in part by the National Science Foundation under award IIS-1702751 and a Graduate Research Fellowship, by a Google Faculty Award, and by the Mani Charitable Foundation.
Authors addresses: A. S. Ross, X. Zhang, and J. Fogarty, Paul G. Allen School of Computer Science & Engineering, DUB Group, University of Washington, Seattle, WA, USA; emails: {ansross, xiaoyiz, jfogarty}@cs.washington.edu; J. O. Wobbrock, The Information School, DUB Group, University of Washington, Seattle, WA, USA; email: wobbrock@uw.edu.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1936-7228/2020/04-ART4 $15.00
https://doi.org/10.1145/3348797

A Large-Scale Longitudinal Analysis of Missing Label Accessibility Failures in Android Apps

Raymond Fok
rayfok@cs.washington.edu
University of Washington
Seattle, Washington, USA

Mingyuan Zhong
myzhong@cs.washington.edu
University of Washington
Seattle, Washington, USA

Anne Spencer Ross
ansross@cs.washington.edu
Bucknell University
Lewisburg, Pennsylvania, USA

James Fogarty
jfogarty@cs.washington.edu
University of Washington
Seattle, Washington, USA

Jacob O. Wobbrock
wobbrock@uw.edu
University of Washington
Seattle, Washington, USA

ABSTRACT

We present the first large-scale longitudinal analysis of missing label accessibility failures in Android apps. We developed a crawler and collected monthly snapshots of 312 apps over 16 months. We use this unique dataset in empirical examinations of accessibility not possible in prior datasets. Key large-scale findings include missing label failures in 55.6% of unique image-based elements, longitudinal improvement in *ImageButton* elements but not in more prevalent *ImageView* elements, that 8.8% of unique screens are unreachable without navigating at least one missing label failure, that app failure rate does not improve with number of downloads, and that effective labeling is neither limited to nor guaranteed by large software organizations. We then examine longitudinal data in individual apps, presenting illustrative examples of accessibility impacts of systematic improvements, incomplete improvements, interface redesigns, and accessibility regressions. We discuss these findings and potential opportunities for tools and practices to improve label-based accessibility.

CCS CONCEPTS

• **Human-centered computing → Accessibility systems and tools**; **Empirical studies in accessibility**.

KEYWORDS

mobile app accessibility; large-scale longitudinal analysis

ACM Reference Format:

Raymond Fok, Mingyuan Zhong, Anne Spencer Ross, James Fogarty, and Jacob O. Wobbrock. 2022. A Large-Scale Longitudinal Analysis of Missing Label Accessibility Failures in Android Apps. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3491102.3502143

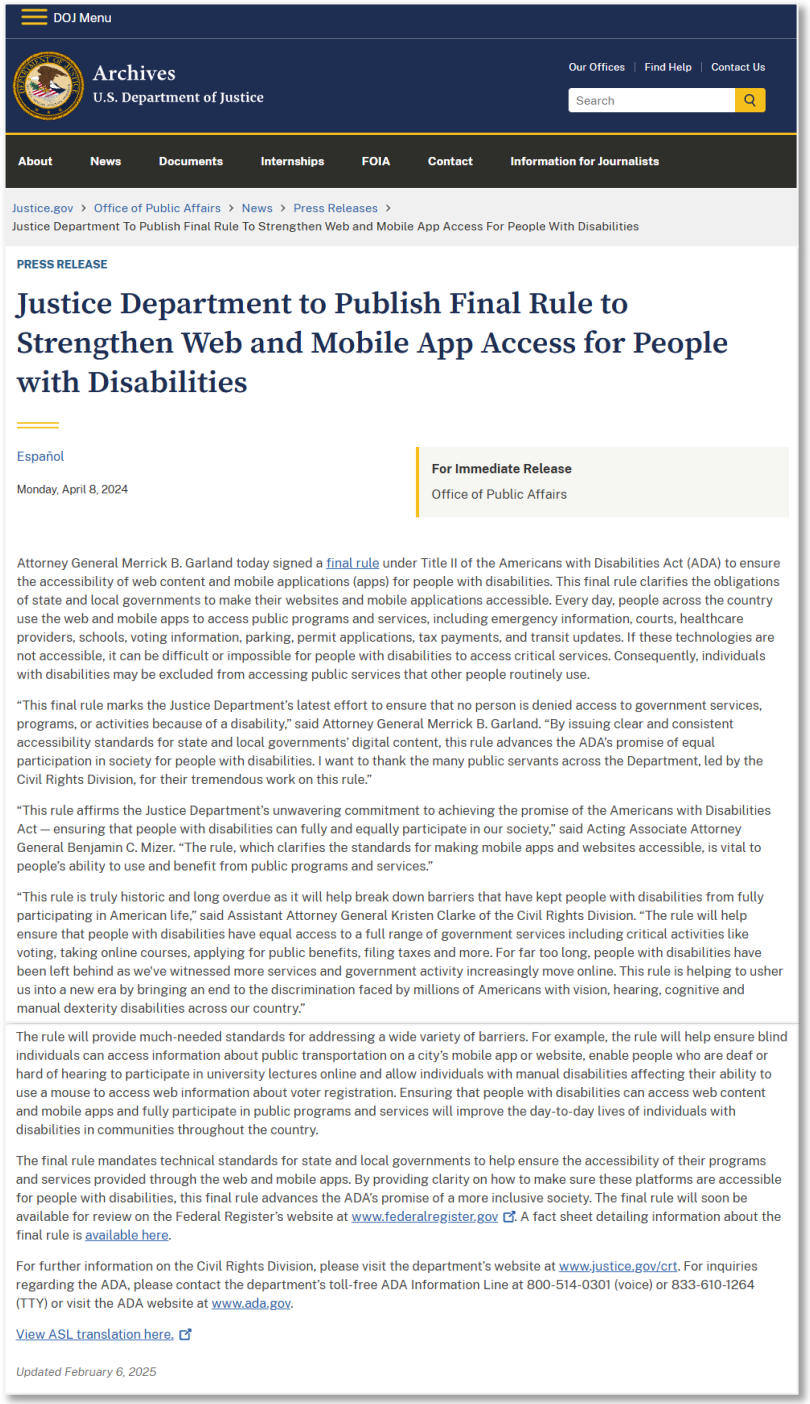
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI '22, April 29-May 5, 2022, New Orleans, LA, USA
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3502143

ASSETS 2017
Best Paper Nominee

ASSETS 2018
Best Paper Nominee

TACCESS 2020

CHI 2022



DOJ Menu

Archives
U.S. Department of Justice

Our Offices | Find Help | Contact Us

Search

About | News | Documents | Internships | FOIA | Contact | Information for Journalists

Justice.gov › Office of Public Affairs › News › Press Releases ›
Justice Department To Publish Final Rule To Strengthen Web and Mobile App Access For People With Disabilities

PRESS RELEASE

Justice Department to Publish Final Rule to Strengthen Web and Mobile App Access for People with Disabilities

Español

Monday, April 8, 2024

For Immediate Release
Office of Public Affairs

Attorney General Merrick B. Garland today signed a final rule under Title II of the Americans with Disabilities Act (ADA) to ensure the accessibility of web content and mobile applications (apps) for people with disabilities. This final rule clarifies the obligations of state and local governments to make their websites and mobile applications accessible. Every day, people across the country use the web and mobile apps to access public programs and services, including emergency information, courts, healthcare providers, schools, voting information, parking, permit applications, tax payments, and transit updates. If these technologies are not accessible, it can be difficult or impossible for people with disabilities to access critical services. Consequently, individuals with disabilities may be excluded from accessing public services that other people routinely use.

"This final rule marks the Justice Department's latest effort to ensure that no person is denied access to government services, programs, or activities because of a disability," said Attorney General Merrick B. Garland. "By issuing clear and consistent accessibility standards for state and local governments' digital content, this rule advances the ADA's promise of equal participation in society for people with disabilities. I want to thank the many public servants across the Department, led by the Civil Rights Division, for their tremendous work on this rule."

"This rule affirms the Justice Department's unwavering commitment to achieving the promise of the Americans with Disabilities Act — ensuring that people with disabilities can fully and equally participate in our society," said Acting Associate Attorney General Benjamin C. Mizer. "The rule, which clarifies the standards for making mobile apps and websites accessible, is vital to people's ability to use and benefit from public programs and services."

"This rule is truly historic and long overdue as it will help break down barriers that have kept people with disabilities from fully participating in American life," said Assistant Attorney General Kristen Clarke of the Civil Rights Division. "The rule will help ensure that people with disabilities have equal access to a full range of government services including critical activities like voting, taking online courses, applying for public benefits, filing taxes and more. For far too long, people with disabilities have been left behind as we've witnessed more services and government activity increasingly move online. This rule is helping to usher us into a new era by bringing an end to the discrimination faced by millions of Americans with vision, hearing, cognitive and manual dexterity disabilities across our country."

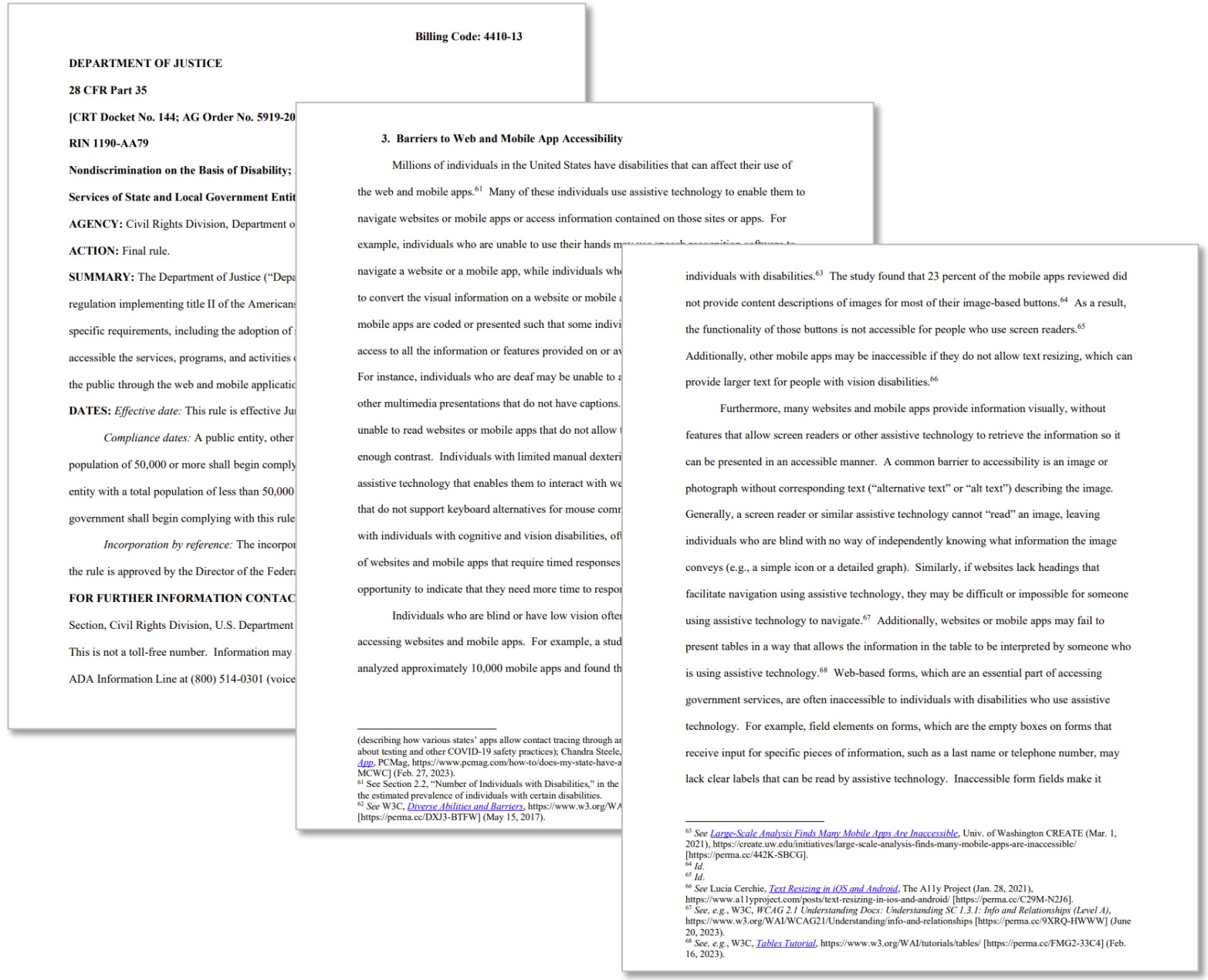
The rule will provide much-needed standards for addressing a wide variety of barriers. For example, the rule will help ensure blind individuals can access information about public transportation on a city's mobile app or website, enable people who are deaf or hard of hearing to participate in university lectures online and allow individuals with manual disabilities affecting their ability to use a mouse to access web information about voter registration. Ensuring that people with disabilities can access web content and mobile apps and fully participate in public programs and services will improve the day-to-day lives of individuals with disabilities in communities throughout the country.

The final rule mandates technical standards for state and local governments to help ensure the accessibility of their programs and services provided through the web and mobile apps. By providing clarity on how to make sure these platforms are accessible for people with disabilities, this final rule advances the ADA's promise of a more inclusive society. The final rule will soon be available for review on the Federal Register's website at www.federalregister.gov. A fact sheet detailing information about the final rule is available here.

For further information on the Civil Rights Division, please visit the department's website at www.justice.gov/crt. For inquiries regarding the ADA, please contact the department's toll-free ADA Information Line at 800-514-0301 (voice) or 833-610-1264 (TTY) or visit the ADA website at www.ada.gov.

View ASL translation here.

Updated February 6, 2025



Billing Code: 4410-13

DEPARTMENT OF JUSTICE

28 CFR Part 35

[CRT Docket No. 144; AG Order No. 5919-20

RIN 1190-AA79

Nondiscrimination on the Basis of Disability;

Services of State and Local Government Entit

AGENCY: Civil Rights Division, Department o

ACTION: Final rule.

SUMMARY: The Department of Justice ("Depa

regulation implementing title II of the American

specific requirements, including the adoption of

accessible the services, programs, and activities

the public through the web and mobile applicatic

DATES: *Effective date:* This rule is effective Ju

Compliance dates: A public entity, other

population of 50,000 or more shall begin comply

entity with a total population of less than 50,000

government shall begin complying with this rule

Incorporation by reference: The incorpor

the rule is approved by the Director of the Federa

FOR FURTHER INFORMATION CONTAC

Section, Civil Rights Division, U.S. Department

This is not a toll-free number. Information may

ADA Information Line at (800) 514-0301 (voice

3. Barriers to Web and Mobile App Accessibility

Millions of individuals in the United States have disabilities that can affect their use of the web and mobile apps.[61] Many of these individuals use assistive technology to enable them to navigate websites or mobile apps or access information contained on those sites or apps. For example, individuals who are unable to use their hands m

navigate a website or a mobile app, while individuals wh

to convert the visual information on a website or mobile

mobile apps are coded or presented such that some indivi

access to all the information or features provided on or a

For instance, individuals who are deaf may be unable to a

other multimedia presentations that do not have captions.

unable to read websites or mobile apps that do not allow

enough contrast. Individuals with limited manual dexteri

assistive technology that enables them to interact with we

that do not support keyboard alternatives for mouse comm

with individuals with cognitive and vision disabilities, of

of websites and mobile apps that require timed responses

opportunity to indicate that they need more time to respo

Individuals who are blind or have low vision ofte

accessing websites and mobile apps. For example, a stud

analyzed approximately 10,000 mobile apps and found th

(describing how various states' apps allow contact tracing through a
about testing and other COVID-19 safety practices); Chandra Steele,
App, PCMag, https://www.pcmag.com/how-to/does-my-state-have-a
MCWC] (Feb. 27, 2023).
[61] See Section 2.2, "Number of Individuals with Disabilities," in the
the estimated prevalence of individuals with certain disabilities.
[62] *See* W3C, *Diverse Abilities and Barriers*, https://www.w3.org/W
[https://perma.cc/DXJ3-BTFW] (May 15, 2017).

individuals with disabilities.[63] The study found that 23 percent of the mobile apps reviewed did not provide content descriptions of images for most of their image-based buttons.[64] As a result, the functionality of those buttons is not accessible for people who use screen readers.[65] Additionally, other mobile apps may be inaccessible if they do not allow text resizing, which can provide larger text for people with vision disabilities.[66]

Furthermore, many websites and mobile apps provide information visually, without features that allow screen readers or other assistive technology to retrieve the information so it can be presented in an accessible manner. A common barrier to accessibility is an image or photograph without corresponding text ("alternative text" or "alt text") describing the image. Generally, a screen reader or similar assistive technology cannot "read" an image, leaving individuals who are blind with no way of independently knowing what information the image conveys (e.g., a simple icon or a detailed graph). Similarly, if websites lack headings that facilitate navigation using assistive technology, they may be difficult or impossible for someone using assistive technology to navigate.[67] Additionally, websites or mobile apps may fail to present tables in a way that allows the information in the table to be interpreted by someone who is using assistive technology.[68] Web-based forms, which are an essential part of accessing government services, are often inaccessible to individuals with disabilities who use assistive technology. For example, field elements on forms, which are the empty boxes on forms that receive input for specific pieces of information, such as a last name or telephone number, may lack clear labels that can be read by assistive technology. Inaccessible form fields make it

[63] *See Large-Scale Analysis Finds Many Mobile Apps Are Inaccessible*, Univ. of Washington CREATE (Mar. 1, 2021), https://create.uw.edu/initiatives/large-scale-analysis-finds-many-mobile-apps-are-inaccessible/ [https://perma.cc/442K-SBCG].
[64] *Id.*
[65] *Id.*
[66] *See* Lucia Cerchie, *Text Resizing in iOS and Android*, The A11y Project (Jan. 28, 2021), https://www.a11yproject.com/posts/text-resizing-in-ios-and-android/ [https://perma.cc/C29M-N2J6].
[67] *See, e.g.*, W3C, *WCAG 2.1 Understanding Docs: Understanding SC 1.3.1: Info and Relationships (Level A)*, https://www.w3.org/WAI/WCAG21/Understanding/info-and-relationships [https://perma.cc/9XRQ-HWWW] (June 20, 2023).
[68] *See, e.g.*, W3C, *Tables Tutorial*, https://www.w3.org/WAI/tutorials/tables/ [https://perma.cc/FMG2-33C4] (Feb. 16, 2023).

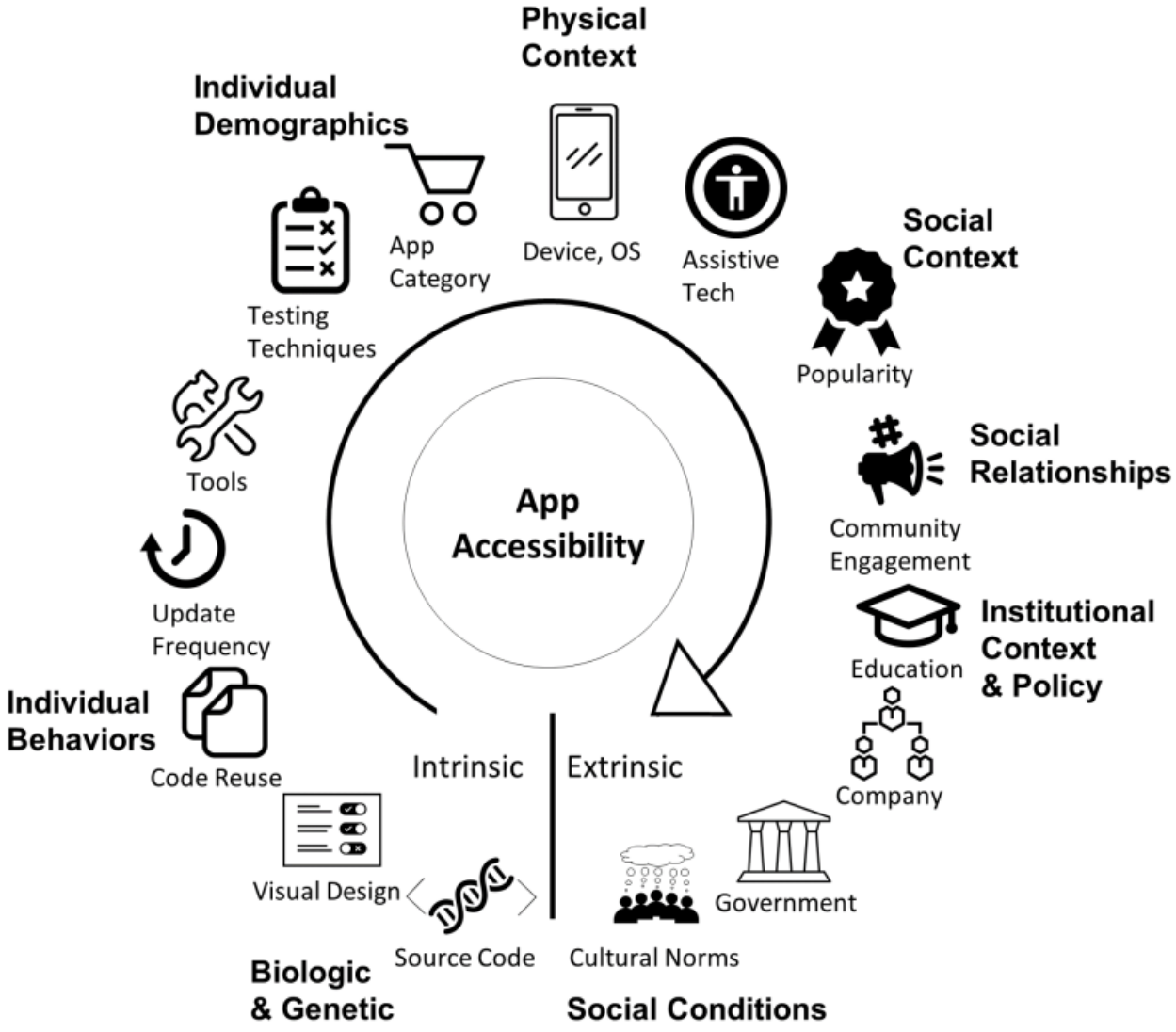
https://www.ada.gov/assets/pdfs/web-rule.pdf (pp. 23-24)

Individuals who are blind or have low vision often confront significant barriers to accessing websites and mobile apps. For example, a study from the University of Washington analyzed approximately 10,000 mobile apps and found that many are highly inaccessible to individuals with disabilities.[63] The study found that 23 percent of the mobile apps reviewed did not provide content descriptions of images for most of their image-based buttons.[64] As a result, the functionality of those buttons is not accessible for people who use screen readers.[65] Additionally, other mobile apps may be inaccessible if they do not allow text resizing, which can provide larger text for people with vision disabilities.[66]

https://www.ada.gov/assets/pdfs/web-rule.pdf (pp. 23-24)

Reflection:
Dive Into Other Disciplines

- Accessibility is not only a technical challenge, it is a human, organizational, social, legal, political, and societal challenge.
- We therefore must be multi-disciplinary to meet this challenge.
- Diving into other disciplines can be difficult and time-consuming, even full of dead-ends.
- But it can also open our minds and vocabularies to seeing and addressing problems in new ways.



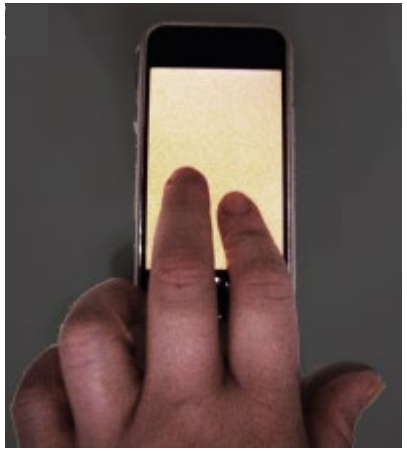
Anne Spencer Ross et al., ASSETS '17

Review



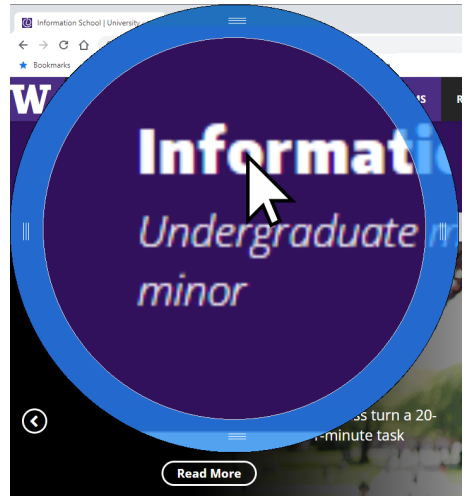
1. Prioritize depth

   EdgeWrite (2001 – 2006)



2. Participant-driven prototyping

   Slide Rule (2007 – 2008)



3. Deploy and support in the real world

   Pointing Magnifier (2010 – present)


contributed articles

DOI:10.1145/3148051

By focusing on users' abilities rather than disabilities, designers can create interactive systems better matched to those abilities.

BY JACOB O. WOBBROCK, KRZYSZTOF Z. GAJOS, SHAUN K. KANE, AND GREGG C. VANDERHEIDEN

Ability-Based Design

RECALL THE LAST time you took a trip out of town. Perhaps you were traveling to a conference far from home. Remember the many forms of transportation you endured: cars, buses, airplanes, and trains. Not only were you responsible for moving yourself over a great distance, you had to move your things as well, including books and baggage. Remember the cramped spaces, sharp elbows, body aches, and exhaustion. Feel again your desire to simply be at your destination with your possessions intact . . .

Such journeys remind us of our physical embodiment in the physical world, that much of our lived experience is fundamentally physical, and that we must contend with the world on physical terms. As computing professionals, we might be tempted to forget this, as our keystrokes summon data instantly from across the globe. But as humans, we still interact with that data through physical devices and displays using our physical senses and bodies. We and the world interact physically.

Civilization's story of technological progress is in no small part the story of an increasingly built physical environment, from the pyramids to roads to skyscrapers to sanitation systems. Much of our energy, collectively and individually, goes into moving and shaping material for such purposes, altering the physical landscape and our movement through it. Some of our most thrilling experiences come by way of changing our bodies' relation to that landscape: bungee jumping, skydiving, scuba diving, and riding a rollercoaster all provide radically new experiences for our bodies in the world.

As designers and builders of interactive systems for human use, we also play a central role in defining people's relationship to and experience of the physical world.[13,16] When we design things, we take mere ideas, things without form, and embody them in the world, whether simple sketches or cardboard mockups. They could be pixels on a screen or functioning digital devices. Regardless of the medium, to design and build things is to embody ideas that are then encountered and

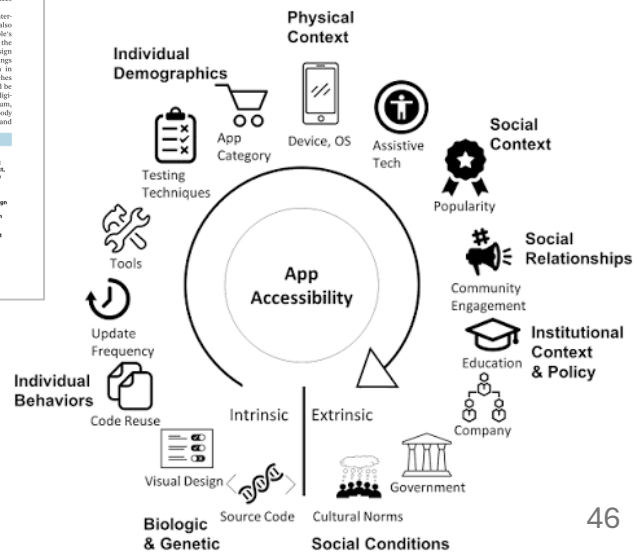
key insights

- Ability-based design is a new design approach for interactive systems that focuses on people's abilities in context, on what people can do, rather than on what they cannot do.
- Ability-based design scrutinizes the "ability assumptions" behind the design of interactive systems, shifting the responsibility of enabling access from users to the system.
- People's abilities may be affected not just by disabilities but by disabling situations; designing for abilities in context leads to more usable, accessible systems for all people.

62 COMMUNICATIONS OF THE ACM | JUNE 2018 | VOL. 61 | NO. 6

4. Pursue conceptual contributions

   Ability-Based Design (2010 – present)



5. Dive into other disciplines

   Epidemiology Model (2016 – 2022)

Promises and Pitfalls of AI.4.A11y

Promises

- A.I. can adapt digital and even physical experiences to each person's abilities, preferences, and situations.
- A.I. products and tools are powerful and increasingly unavoidable.
- A.I. writes code that accelerates the creation of prototypes, products and services.

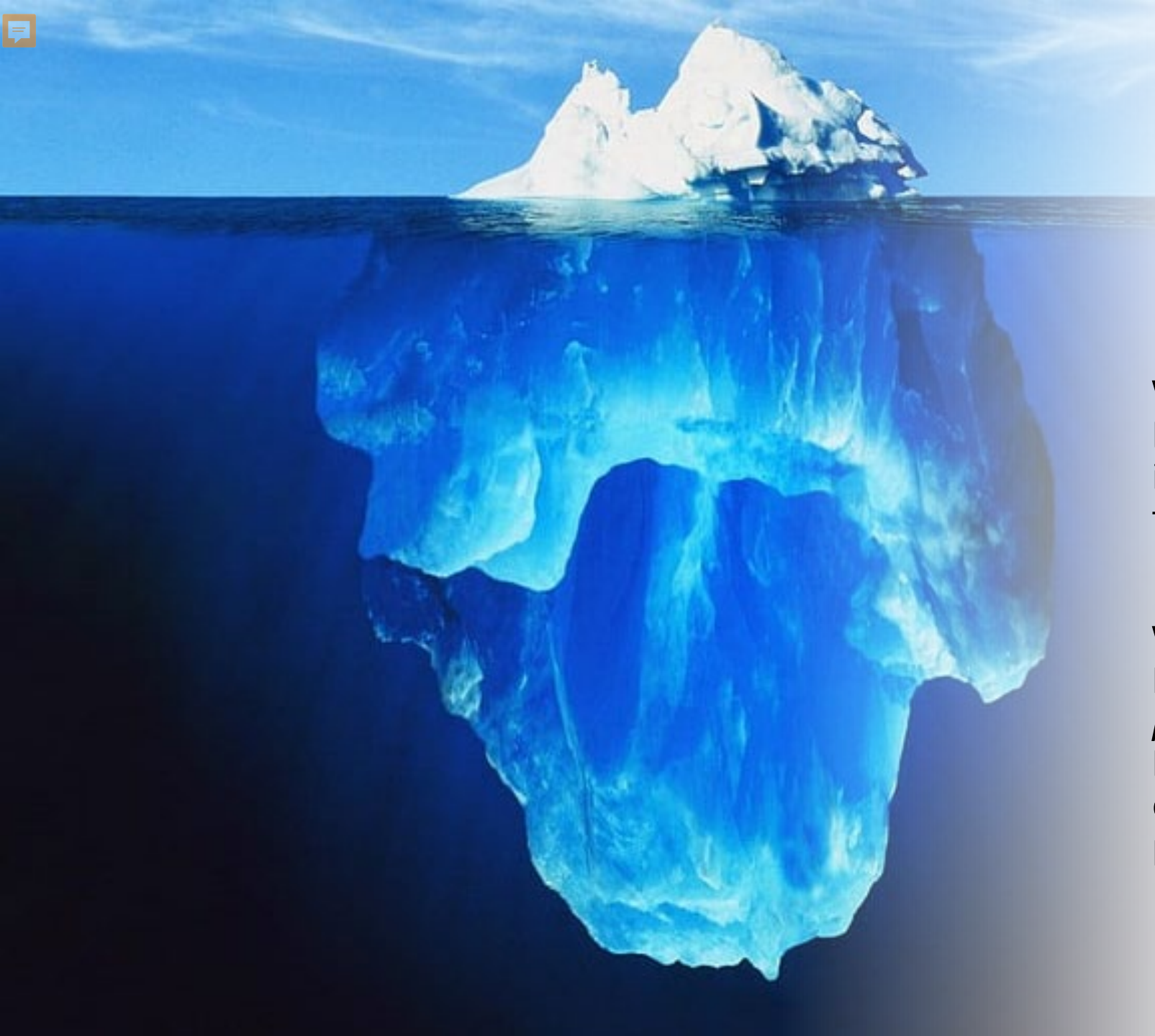
Pitfalls

- People with disabilities must give up far more personal information than others to reap the benefits of A.I.
- A.I. products *themselves* are inaccessible, so we just perpetuate more barriers.
- A.I. writes code that makes *inaccessible* prototypes, products and services.



Reflecting… *forward*?

Can our reflections help us navigate the **future** of A.I. for accessibility?



Prioritize depth

We need to **deeply understand** how people with disabilities interact with A.I., especially if they come to rely on it.

We need to **deeply understand** how A.I. *works with and for people*. What assumptions, biases, and misconceptions does A.I. embody concerning people with disabilities?

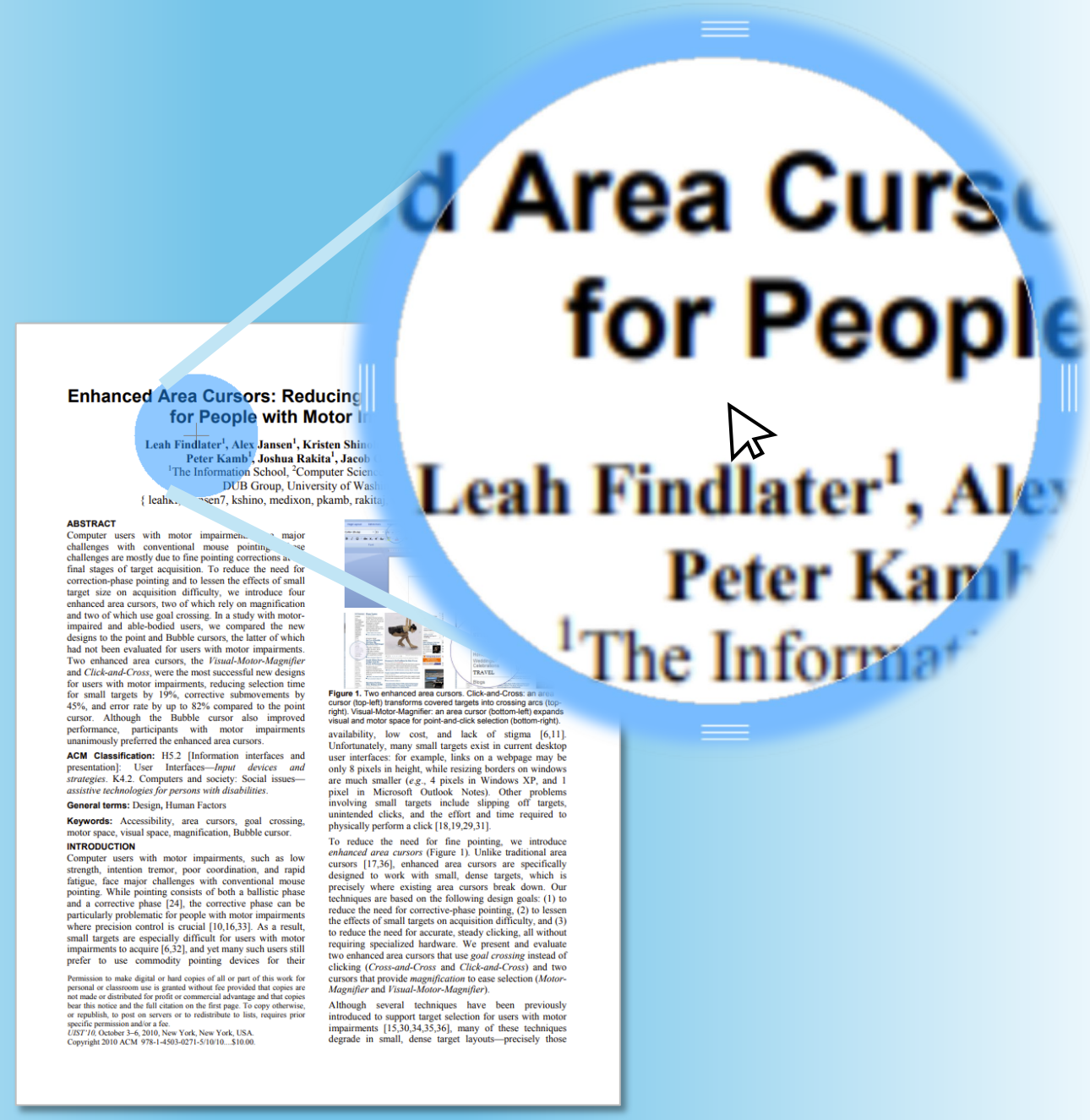


Participant-driven prototyping

A.I.-based products and services are unlikely to be accessible by default.

We need people with disabilities to **drive the creation** of A.I.-based products and services.

Richard Ladner's *Design for User Empowerment* (2015) offers one compelling approach.



Deploy and support in the real world

Accessibility must be approached as an **in-the-world** concept, not a lab-only concept.

Unless we **deploy and support** A.I.-based systems, we will never really know how they affect people with disabilities, and how people with disabilities use (or avoid) them, and why.

Unforeseen or unintended consequences can arise when deploying, so we must **deploy with care**.

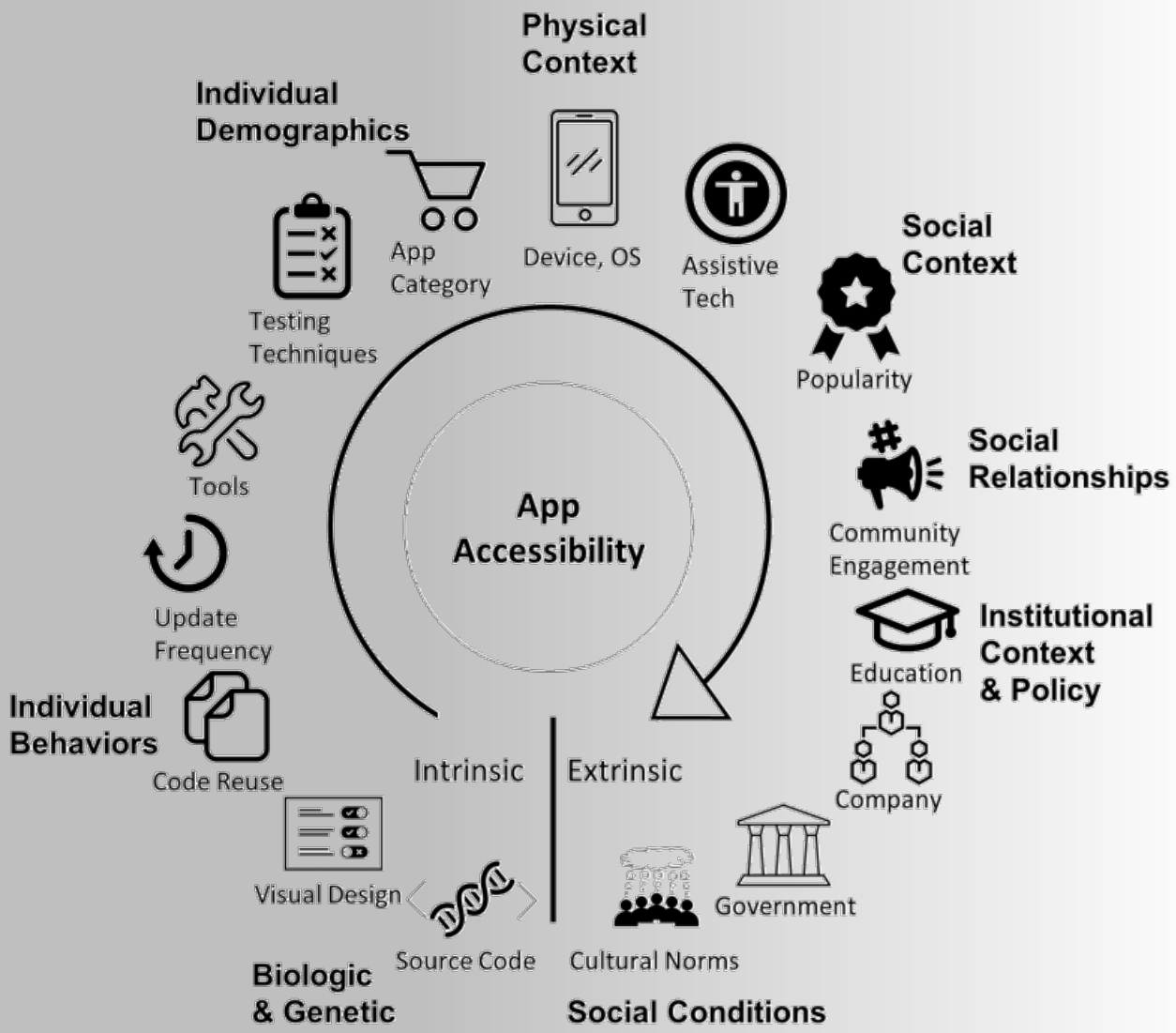


Pursue conceptual contributions

We need better ways of **thinking about** human-A.I. systems in general, and A.I. for accessibility in particular.

What is our **goal vocabulary?** To: augment? amplify? ameliorate? replace? restore? recover? enable? equip? empower?

We need better mental models, design methods, and shared understandings of how A.I. and people **should** work together.



Dive into other disciplines

Accessibility is multi-disciplinary and multi-faceted. It requires many forms of expertise.

To understand the impacts of A.I. on people with disabilities, we need a **holistic approach**.

What expertise do we need to **get ahead** of A.I. rather than to merely **react** to its consequences?









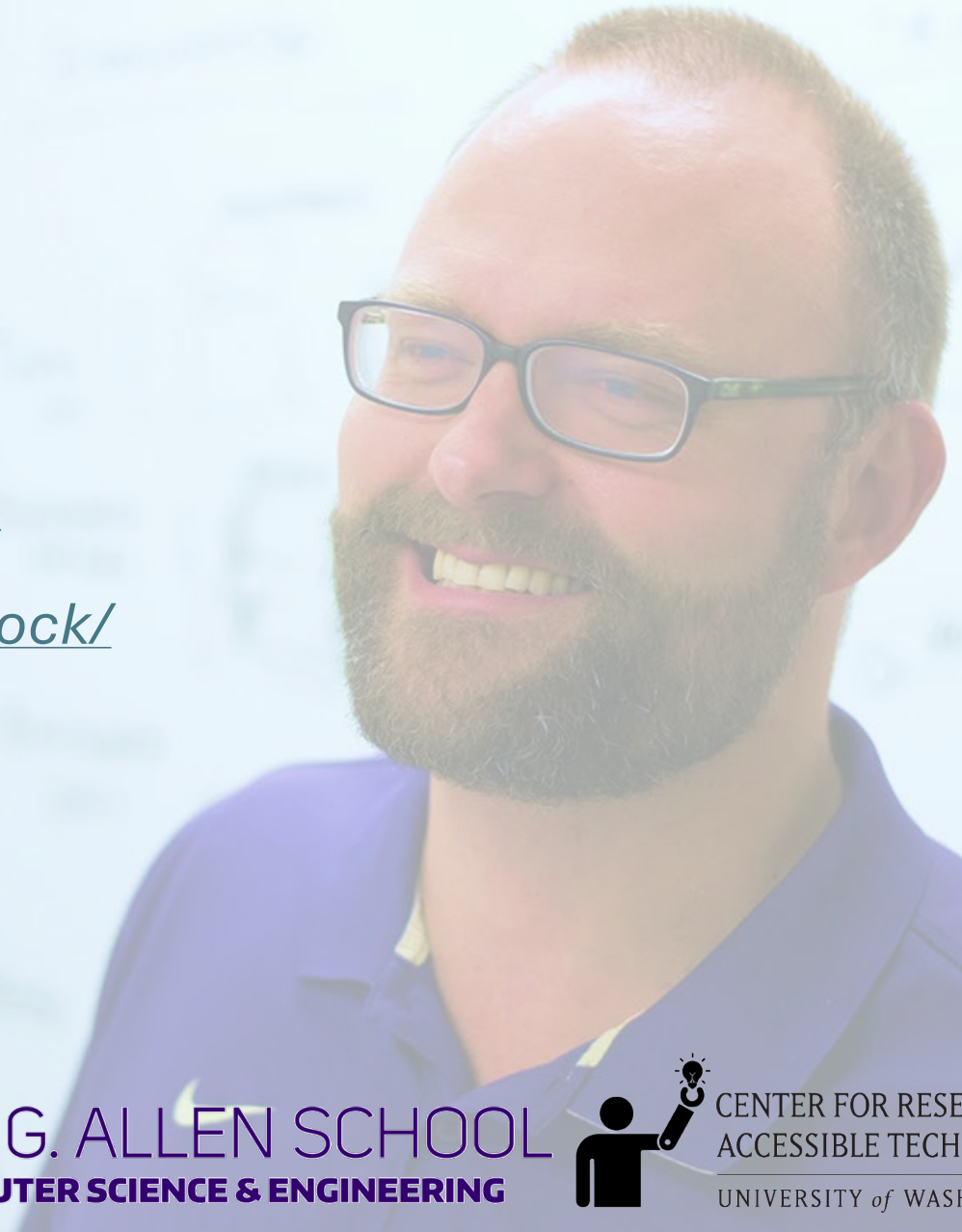
Special Thanks

DESIGN USE BUILD

ACE Lab

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

https://depts.washington.edu/acelab/
https://faculty.washington.edu/wobbrock/



Information School
UNIVERSITY of WASHINGTON

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

CENTER FOR RESEARCH AND EDUCATION ON ACCESSIBLE TECHNOLOGY AND EXPERIENCES
UNIVERSITY of WASHINGTON