Ability-Based Design: Concept, Principles and Examples

JACOB O. WOBBROCK, SHAUN K. KANE
The Information School, University of Washington

KRZYSZTOF Z. GAJOS
Harvard School of Engineering and Applied Sciences

SUSUMU HARADA[1], JON FROEHLICH
Department of Computer Science & Engineering,
University of Washington

Current approaches to accessible computing share a common goal of making technology accessible to users with disabilities. Perhaps because of this goal, they may also share a tendency to centralize disability rather than ability. We present a refinement to these approaches called *ability-based design* that consists of focusing on ability throughout the design process in an effort to create systems that leverage the full range of human potential. Just as user-centered design shifted the focus of interactive system design from systems to users, ability-based design attempts to shift the focus of accessible design from disability to ability. Although prior approaches to accessible computing may consider users' abilities to some extent, ability-based design makes ability its central focus. We offer seven ability-based design principles and describe the projects that inspired their formulation. We also present a research agenda for ability-based design.

1. INTRODUCTION

Despite more than twenty years of research in accessible computing, user interfaces still pose access challenges for many people with disabilities. Ironically, a consistent centralization of *disability* may be partly to blame.

Authors' addresses: J. O. Wobbrock, Information School, University of Washington, Seattle, Washington, USA 98195-2840. E-mail: wobbrock@uw.edu; S. K. Kane, E-mail: skane@uw.edu; K. Z. Gajos, Harvard School of Engineering and Applied Sciences, Cambridge, Massachusetts, USA 02138. E-mail: kgajos@eecs.harvard.edu; S. Harada, 1623-14 Shimotsuruma, Yamato, Kanagawa, 242-8502, Japan. E-mail: harada@gmail.com; Jon Froehlich, E-mail: jfroehli@cs.washington.edu.

---

[1]Currently employed at IBM Tokyo Research Laboratory.

By its very name, a *dis*-ability is not so much a thing as the lack of a thing, namely the lack of ability. Taking this perspective, a person with a disability is *un*-able to perform certain everyday tasks, and requires assistance to compensate for his or her limitation. However, people do not have dis-abilities any more than they have dis-money or dis-height. Abilities vary across a range, from those of Olympic athletes (Figure 1) to toddlers and the elderly. Likewise, ability is not static; it is influenced by the context in which it is exercised (Newell 1995). With this in mind, the appropriate question when designing accessible technologies is not, "What disability does a person have?" but rather, "What can a person *do*?" (Chickowski 2004). This question prompts a refocusing of accessible computing from disabilities to abilities, much as user-centered design refocused interactive system development from systems to users (Gould and Lewis 1985). We call this refocused perspective *ability-based design*.

Figure 1. The logo on the left is commonly associated with *dis*-ability, which is focused on something lacking. Contrast this with the logo on the right from the National Veterans Wheelchair Games (http://www.wheelchairgames.org), which communicates *ability*: strength, speed, power, and determination.

In making the shift to ability-based design, we move away from assisting human users to conform to inflexible computer systems, and instead consider how systems can be made to fit the abilities of whoever uses them. As an example, consider a computer user with limited dexterity. This user might have difficulty using a mouse to click on targets that were designed for people with "average" dexterity. Currently, this user must struggle to use the interface as-is, employ a built-in or add-on software-based accessibility aid (if one is available), or choose to purchase a specialized input device that is designed for people with disabilities. An ability-based design approach would instead provide a system that is aware of the abilities of the user and provides an interface better suited to those abilities. An example of this approach is our SUPPLE system (Gajos

et al. 2010), which measures the user's pointing abilities and automatically redesigns, rearranges, and resizes the interface to maximize performance. SUPPLE is discussed in section 5.8.

In ability-based design, the challenge is to identify abilities in a reliable fashion and to design technologies that take advantage of users' abilities to interact with available hardware and software resources. Adaptation is not required for ability-based design, but adaptation can move the burden of conforming from the human user to the system.

We find that ability-based design is a useful refinement to existing accessible computing approaches such as rehabilitation engineering, universal design, and inclusive design. Although prior approaches consider users' abilities to some extent, ability-based design makes ability its central focus.

The goals of this article are to describe ability-based design, put forward its principles, and discuss projects that inspired its formulation. This article also articulates research challenges for furthering ability-based design. It is the authors' hope to advance the conversation surrounding "people with disabilities" to one that primarily considers "people with abilities" in their contexts of use.

2. TIMELINESS OF ABILITY-BASED DESIGN

Ability-based design is timely, as recent work has explored methods for capturing, measuring, and modeling the abilities of diverse users (Casali 1995, Gajos *et al.* 2007, Gajos *et al.* 2008, Hurst *et al.* 2007, Hurst *et al.* 2008a, Keates *et al.* 2002, Law *et al.* 2005, Price and Sears 2008). Capturing, measuring, and modeling abilities are difficult feats, as large variations in human ability present numerous challenges to sensing, inference, abstraction, and measurement. However, the malleability of software and its potential for sensing and adaptation suggest that if abilities *can* be modeled, they may be accommodated by ability-based user interfaces (Gajos *et al.* 2008).

It is important to distinguish ability-based design's concepts of ability capture, measurement, and modeling from both clinical functional assessment and formal disability characterization. Clinical functional assessment requires a clinician working from a medical or occupational perspective to measure a person's abilities with respect to job performance

in the presence of a condition, disease, or injury (Gross 2004, Matheson 2004, Pransky and Dempsey 2004). Disability characterization, *e.g.*, using the World Health Organization's International Classification of Functioning, Disability and Health (ICF) (World Health Organization 2009), describes disability in light of one's body, activity, and environment. Both functional assessment and disability characterization usually produce outcomes at too high a level for informing user interface optimizations at either design-time or run-time.

Although ability-based design can be achieved without automatic adaptation, recent advances in adaptive user interfaces offer potential for advancing ability-based design. Achievements in automatic user interface generation (Gajos *et al.* 2010), ephemeral adaptation (Findlater *et al.* 2009), adaptation to different user skills (Hurst *et al.* 2007), and adaptation to changing user contexts (Kane *et al.* 2008c) demonstrate that interactive technologies can detect and adapt to a user's abilities, and therefore support an ability-based design approach.

Yet another reason for the timeliness of ability-based design is that commodity computing peripherals are cheaper and more widely available than ever before. Although many people with disabilities are unable to use such peripherals "off-the shelf," devices may be made usable through software that accommodates users' abilities. This approach contrasts with the more traditional approach of creating specialized hardware to enable people with disabilities to access unadapted software. Prior research shows that the cost, complexity, configuration, and maintenance of specialized hardware and software are perpetual barriers to access (Dawe 2004), and that abandonment rates are high for specialized systems (Bates and Istance 2003, Goette 1998, Koester 2003). Additional work shows that no more than 60% of people who indicate a need for access technologies actually use them (Fichten *et al.* 2000). Many of the above barriers can be circumvented by designing software for use with commodity input devices like mice, touchpads, and trackballs, placing the burden of making these devices effective on the software with which they are used. Although the abilities of some people with severe impairments may not be sufficient to use commodity input devices, many people with mild to moderate impairments have the ability to manipulate such devices. The projects that

inspired ability-based design (see section 5) demonstrate this approach, as do prior studies by others in which people with disabilities used commodity input devices with noteworthy success (Casali 1992). Of course, for people who need or prefer to use specialized devices, the goal of ability-based design is to provide interfaces that optimally take advantage of the combination of these devices' strengths and the abilities of the people using them.

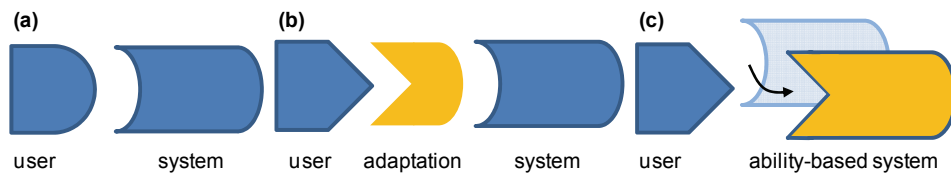## 3. PRIOR APPROACHES TO ACCESSIBLE COMPUTING

Prior approaches to accessible computing are numerous, unavoidably and appropriately having considerable overlap with each other and with ability-based design. We position ability-based design as a refinement to and refocusing of prior approaches that emphasizes the importance of considering human abilities during the design process. All accessible computing approaches share a common goal of improving independence, access, and quality of life for people with disabilities. Where each approach directs a designer's or researcher's focus, however, may differ.

When considering how a technology can be modified to suit a particular user, we consider the degree of its *adaptivity* and/or *adaptability* (Findlater and McGrenere 2004, Stephanidis *et al.* 1995). By "adaptivity" we mean the degree to which software can change itself in response to user behavior. By "adaptability" we mean the degree to which software can be customized by a user, therapist, or caregiver. The following survey of approaches will be regarded along these dimensions.

3.1 Assistive Technology

Assistive technology is a term that covers a broad set of technologies, methods, and views (Cook and Hussey 2002, Vanderheiden 1998). The approach developed out of World War II and the postwar era, and thus emerged prior to the proliferation of interactive computing. Assistive technology therefore has a tendency to assume that the environment is immutable, like a physical product or building, and cannot be easily changed. Thus, a focus of assistive technology is largely that of fitting "non-standard users" to standard technology by means of an assistive component, often an add-on inserted between the user and the system. Figure 2 illustrates this approach with symbols adapted from prior work

(Edwards 1995).

Figure 2. (a) A user whose abilities match those presumed by the system. **(b)** A user whose abilities do not match those presumed by the system. Because the system is inflexible, the user must be adapted to it. **(c)** An ability-based system is designed to accommodate the user's abilities. It may adapt or be adapted to them. Our symbols are based on those from prior work (Edwards 1995).

One might reasonably argue that even in Figure 2b, a system is being adapted to a user's abilities. After all, the user is not being changed, *per se*; rather, a component is being inserted between the user and system, which could be viewed as modifying the existing system, not unlike the ability-based system in Figure 2c. We disagree with this view for a few reasons. First, the inserted component in Figure 2b is rarely part of the original system, and thus requires procurement. Second, the burden of change in Figure 2b resides with the user, not the system; the user must decide how he or she can become amenable to the system, and choose an adaptation accordingly. Meanwhile, the system in Figure 2b remains oblivious to the user's abilities, and does not change from one user to the next. Take, for example, a user who employs a mouth stick to type on a keyboard. The user knows what action the keyboard requires, but the keyboard knows nothing of the user's abilities. The mouth stick is inherent neither to the user nor to the system, but is placed between the user and system to make the user "acceptable" to the demands of the inflexible system.

As a result of its use of "inserted" or "add-on" devices, assistive technology is sometimes criticized for resulting in "separate but equal" solutions for people with disabilities (Hazard 2008, Steinfeld 1994, Stephanidis 2001a). Assistive add-ons can also stigmatize a person who would rather not be seen using "special" or "other" technology (Shinohara and Tenenberg 2007, Shinohara and Wobbrock 2011). By contrast, ability-based design attempts to shift the burden of accommodation from the human to the system (Figure 2c).

Adaptation plays an important role in ability-based design. Ability-based

design advocates personalized user interfaces that adapt themselves or are easily adapted by the human user. This removes the need for a clinician to set up the technology, scales to the masses, and is inexpensive. Moreover, user interfaces can adapt or be made adaptable to the changing skills of users. Unlike a cane, unpowered wheelchair, or building ramp, computer technology can observe users' performance, model it, and use those models to predict future performance, adapting or making suggestions for adaptations (Gajos and Weld 2006, Gajos *et al.* 2007, Gajos *et al.* 2008, Stephanidis 2001a). If automatic adaptation is unwarranted or undesirable, software can still provide its users with options or suggestions for customization.

3.2 Rehabilitation Engineering

Rehabilitation engineering is an engineering approach, and as such, has sought to quantify, measure, and track human performance for the sake of providing better-fitting adaptations (Smith and Leslie 1990). Rehabilitation engineering was created in part as a response to the trial-and-error approaches of many assistive technology practitioners (Kondraske 1988b). It is no surprise, then, that rehabilitation engineering and ability-based design both share a commitment to understanding users' performance. However, the focus of rehabilitation engineering is much broader than just computer use. Rehabilitation engineering focuses on developing engineering models of human performance (Kondraske 1988a, Kondraske 1995, Persad *et al.* 2007) and often measures that performance on task batteries that include many non-computer-related tasks (Kondraske 1990a, Kondraske 1990b, Smith and Kondraske 1987). Also, accommodations in rehabilitation engineering are often custom add-on devices or machines, giving rehabilitation engineering much in common with assistive technology.

3.3 Universal Design

Universal design is a set of design principles that grew out of architecture (Mace *et al.* 1991, Steinfeld 1994, Story 1998), partly as a response to the limitations of add-on approaches like assistive technology and rehabilitation engineering (Vanderheiden 1998). Its founders were mainly concerned with physical spaces and physical tools, although they

crafted general principles applicable to many areas of design. Universal design inescapably has a "one size fits all" ring, which suits door handles, knives, and building entrances, but may be more difficult to employ with interactive computer systems, which deeply and for prolonged periods engage users' motor, sensory, cognitive, and affective faculties.

3.4 Universal Usability

Universal usability provides guidelines for designing interfaces that are usable by the widest range of people possible (Lazar 2007, Shneiderman 2000, Vanderheiden 2000). Universal usability therefore does not make special provision for people with disabilities (Vanderheiden 2000). Universal usability is equally concerned with disparities of access and use owing to gender, economics, literacy, age, culture, and so forth. Like universal design, universal usability inevitably hints at a "one size fits all" ideal. It also places special emphasis on "bridg[ing] the gap between what users *know* and what they *need to know*" (Shneiderman 2000). Ability-based design could be considered one approach that might help further the grand vision of universal usability.

3.5 Design for All

Design for all hails from continental Europe and is focused on achieving barrier-free access to "the information society" for people with disabilities and the elderly (Stephanidis *et al.* 1998). As a concept, it is very similar to universal design. However, it has been observed that the question asked by the so-called "universal" approaches like universal design, universal usability, and design for all is, "what can everyone do?" (Harper 2007) Ability-based design asks a much more targeted question: "what can *you* do?" The problematic nature of the first question has been recently observed (Harper 2007), and the suggestion has been made to move from *design-for-all* to *design-for-one* (Ringbauer *et al.* 2007). Ability-based design may be a viable foundation for achieving design-for-one.

3.6 User Interfaces for All

Conceptually based on design for all, *user interfaces for all (UI4All)* has emerged specifically to describe software principles for accessible

computing (Stary 1997, Stephanidis *et al.* 1995, Stephanidis *et al.* 1997, Stephanidis 2001b). UI4All advocates *unified user interfaces* as a software approach to providing customized user interfaces. In UI4All, abstract user interface representations are mapped to one of many concrete user interface templates, possibly at run-time. This approach is entirely compatible with ability-based design provided that users' abilities, and not only, *e.g.*, their preferences, demographics, and/or prior settings, inform the mapping of abstract representations to concrete interface templates.

3.7 Inclusive Design

Hailing primarily from the United Kingdom, *inclusive design* focuses on factors that cause "design exclusion" (Keates et al. 2000, Keates and Clarkson 2003, Keates and Clarkson 2004, Newell and Gregor 2000). Design exclusion arises in the presence of barriers to access as designers unintentionally create products or services that exclude users because of subconscious biases and assumptions about users' abilities. It would be impossible for one designer to retain the necessary awareness of exclusion caused by every choice he or she makes. In fact, it is likely that to carry such a mental load would adversely affect a designer's ability to work. This is one reason why design itself cannot scale to address the needs of every individual with abilities different than the societal "average." On the other hand, ability-based systems that can observe and accommodate users directly have a much better chance at scaling (Gajos *et al.* 2007).

3.8 Extra-Ordinary Human-Computer Interaction

Finally, we find the idea of *extra-ordinary human-computer interaction* (Newell 1995) to be similar to ability-based design in that it recognizes that all users have some abilities, and that some users have *extra*-ordinary abilities. This approach explicitly debunks the myth of the "average user" (Edwards 1995, Newell 1995), and acknowledges, but does not quantify, the effects of context on user performance. Extra-ordinary human-computer interaction provides a perspective on the relationship among ability, technology, and context, and provides a useful conceptual framework in which to situate ability-based design.

An important feature of extra-ordinary human-computer interaction is its acknowledgement that *context* can temporarily reduce a user's abilities in

ways similar to the effects of personal health-related impairments (Newell 1995). This insight underpins recent attempts to describe, measure, and accommodate *situationally-induced impairments and disabilities*, or "situational impairments" for short (Kane *et al.* 2008c, Lin *et al.* 2007, Sears *et al.* 2003, Sears *et al.* 2008, Wobbrock 2006, Yamabe and Takahashi 2007, Yesilada *et al.* 2010). Situational impairments arise when aspects of a user's environment adversely affect his or her ability to perform certain activities. Examples of contextual factors that may impact abilities are ambient noise, distraction, diverted attention, body motion, walking vibration, weather (*e.g.*, rain water, cold temperatures), restrictive clothing (*e.g.*, gloves causing "fat fingers"), uneven terrain (*e.g.*, stairs), glare, low light or darkness, encumbering baggage, confined or crowded spaces, and so on. Designs that benefit people with disabilities may also benefit nondisabled people who experience temporary situational impairments. In both cases, ability-based design focuses on what users can do, and in what contexts they can do it.

3.9 Other Ability-Based Approaches

Papers that explicitly centralize "ability" in design are surprisingly few. One recent example discusses "ability-centered design" (Evenson *et al.* 2010), which is a perspective combining notions of ability, universality, adaptivity, lifelikeness, static and dynamic objects, and complex systems. This article is more of a reflection than a prescribed design method, and its intention is to provoke thought and discussion. Another recent example is an article on "ability-based systems design" (Jipp *et al.* 2008b), which offers an abstract mathematical description of systems that detect performance in everyday tasks and adapt to that performance. The article addresses only cognitive abilities and makes no mention of people with disabilities. A follow-up piece discusses modelling cognitive capability using Bayesian networks (Jipp *et al.* 2008a), while later follow-ups discuss modelling human performance with wheelchairs (Jipp *et al.* 2009a, Jipp *et al.* 2009b). Lastly, our own prior work on SUPPLE refers to "ability-based interfaces" as those being generated from a cost-optimization procedure parameterized by measures of users' abilities from a test battery involving simple mouse-based tasks (Gajos *et al.* 2007, Gajos *et al.* 2008).

4. PRINCIPLES OF ABILITY-BASED DESIGN

A series of projects (see section 5) informed the creation of seven principles of ability-based design (Table 1). The first two principles relate to the designer's STANCE, which informs a designer's philosophy, orientation, focus, reflection, and goals (Kelley and Hartfield 1996). These two principles (1-2) are required for any ability-based design—they constitute the essential refocusing from disability to ability. The next two principles (3-4) concern the INTERFACE, while the last three (5-7) concern ability-based SYSTEMS in general. Principles 3-6 are "recommended," as they enhance any ability-based design but are not required by all designs. Principle 7 is "encouraged," indicating that designers do well to consider upholding this principle when possible, but acknowledging that some ability-based designs may utilize specialized hardware or software, especially for users with severe disabilities. We note that others have discussed many of these or similar principles before. Our contribution here is to bring these ideas under one conceptual roof, and to apply them systematically to a series of technologies.

Seven Principles of Ability-Based Design

STANCE	**1. Ability.**	Designers will focus on ability not *dis*-ability, striving to leverage all that users *can* do.	*Required*
	2. Accountability.	Designers will respond to poor performance by changing systems, not users, leaving users as they are.	*Required*
INTERFACE	**3. Adaptation.**	Interfaces may be self-adaptive or user-adaptable to provide the best possible match to users' abilities.	*Recommended*
	4. Transparency.	Interfaces may give users awareness of adaptations and the means to inspect, override, discard, revert, store, retrieve, preview, and test those adaptations.	*Recommended*
SYSTEM	**5. Performance.**	Systems may regard users' performance, and may monitor, measure, model, or predict that performance.	*Recommended*
	6. Context.	Systems may proactively sense context and anticipate its effects on users' abilities.	*Recommended*
	7. Commodity.	Systems may comprise low-cost, inexpensive, readily available commodity hardware and software.	*Encouraged*

Table 1. Seven principles of ability-based design. Principles are divided into three categories: STANCE, INTERFACE, and SYSTEM. All ability-based designs will uphold principles 1-2. The next four principles (3-6) are recommended, but are not required in every ability-based design. Finally, designers are encouraged to consider upholding the seventh principle, but with the understanding that custom, specialized solutions are often warranted, especially for severely disabled users.

In STANCE (principles 1-2), ability-based designers orient to "what a

person *can* do," and away from both "what a person cannot do" and "what 'everyone' can do." Ability-based design assumes an iterative design process, and adopts an attitude of *accountability* (2) that places the burden of change on the system, not the user. Requiring the user to retrain themselves, or to purchase or use special adaptations, holds the user accountable for problems that should be addressed by the system designer, and may erect the usual barriers of cost, complexity, configuration, and maintenance (Dawe 2004). In the ideal case, an ability-based system should be flexible enough to enable people to use a system without requiring them to alter their bodies, knowledge, or behavior.

The INTERFACE (principles 3-4) deserves special attention in ability-based design because it is there that a user's abilities are exercised and have their effect. The *adaptation* principle (3) refers to interfaces that self-adjust or can be adjusted, often (but not exclusively) in response to performance or context. Adaptation is what, in the best cases, removes the need for assistive add-ons and forces systems to accommodate users, not the other way around (see Figure 2). Adaptation strikes a philosophical chord in supporting users' personal dignity in enabling them to remain as they are, rather than allowing technology to insist that people change to meet *its* needs (Friedman *et al.* 2006). The *transparency* principle (4) furthers this dignity by always enabling users to inspect, override, discard, revert, store, retrieve, preview, and test adaptations. The need for transparency is not limited to self-adaptive systems, and is also necessary for user-adaptable systems. That way, after time has passed or if changes were made unbeknownst to a user, all adaptations remain visible and changeable.

The SYSTEM principles (5-7) of *performance* (5) and *context* (6) both relate to systems having regard for users' actions, possibly at runtime, with underlying support for sensing, monitoring, measuring, modeling, and predicting those actions. While not all ability-based systems will create and maintain a model of a user's performance, these two recommended principles will be upheld by many successful ability-based designs. Regarding the sensing of context, a great deal of future work is necessary to devise both hardware and software sensors for all aspects of relevant context, and to adjust system parameters accordingly. Although

context ultimately affects performance, we choose to cite context as its own principle to emphasize the need for proactive sensing and anticipation of changes in users' performance due to changes in context.

In the *commodity* principle (7), we encourage designers to use, whenever possible, cheap, readily available, off-the-shelf hardware and software, as doing so inevitably lowers the barriers to access caused by cost, complexity, configuration, and maintenance (Dawe 2004), and makes system components more easily replaced. Using commodity devices also allows accessible ability-based software to be disseminated via the Web without the challenge of manufacturing or distributing specialized hardware. As an "encouraged" principle, *commodity* acknowledges that many ability-based systems will have custom components, and that specialized systems tailored for specific users, particularly those with severe disabilities, are often warranted and may indeed be ability-based. Ultimately, for a design to be considered ability-based, it must uphold the first two principles of *ability* and *accountability*; the remaining principles, when appropriately applied, can significantly improve a design's usability and accessibility.

5. PROJECTS INFORMING ABILITY-BASED DESIGN

To illustrate how the principles of ability-based design may be upheld in specific technologies, we present a selection of projects that informed and inspired the formulation of ability-based design. These projects were undertaken prior to the present conceptual work on ability-based design, and accordingly, embody different principles to different extents.

Table 2 summarizes the projects reviewed in this section and the ability-based design principles upheld. Accordingly, it serves as a guide to the remainder of section 5. The required ability-based design principles of *ability* (1) and *accountability* (2) refer to the designers' stance, not to a designed artifact, *per se*. It will be clear that principles 1 and 2 can be fairly assumed for all projects in Table 2. These principles are therefore omitted in each project summary, below.

5.1 Dynamic Keyboard Model

Trewin and Pain analyzed typing performance by people with motor impairments and developed a system to dynamically model users' typing

Example Projects and Principles

Desktop Text Entry	**5.1 Dynamic Keyboard Model** (Trewin and Pain 1997)	keyboard typing for users with motor impairments	ability, accountability, adaptation, transparency, performance, commodity
	5.2 Invisible Keyguard (Trewin 2002)	keyboard typing for users with motor impairments	ability, accountability, adaptation, performance, commodity
	5.3 Input Device Agent (Koester *et al.* 2005, Koester *et al.* 2007a)	keyboard typing and mouse pointing for users with motor impairments	ability, accountability, adaptation, performance, commodity
	5.4 TrueKeys (Kane *et al.* 2008b)	keyboard typing for users with motor impairments	ability, accountability, transparency, performance, commodity
	5.5 Trackball EdgeWrite (Wobbrock and Myers 2006a, Wobbrock and Myers 2006c, Wobbrock and Myers 2007)	gestural text entry from trackballs for people with spinal cord injuries	ability, accountability, adaptation, transparency, performance, commodity
Mouse Pointing	**5.6 Steady Clicks** (Trewin *et al.* 2006)	mouse pointing for users with motor impairments	ability, accountability, performance, commodity
	5.7 Angle Mouse (Wobbrock *et al.* 2009)	mouse pointing for users with motor impairments	ability, accountability, adaptation, transparency, performance, commodity
	5.8 SUPPLE (Gajos *et al.* 2007, Gajos *et al.* 2008, Gajos *et al.* 2010)	mouse pointing for users with motor impairments	ability, accountability, adaptation, transparency, performance, commodity
	5.9 Automatic mouse pointing assessment (Hurst *et al.* 2007, Hurst *et al.* 2008a)	automatic assessment of mouse pointing performance	ability, accountability, adaptation, performance, commodity
	5.10 VoiceDraw (Harada *et al.* 2007)	paintbrush control for users with motor impairments	ability, accountability, adaptation, performance, commodity
Mobile Devices	**5.11 Barrier pointing** (Froehlich *et al.* 2007)	stylus pointing for users with motor impairments	ability, accountability, performance, commodity
	5.12 Walking user interfaces (Kane *et al.* 2008c)	touch screen access for walking users	ability, accountability, adaptation, context, commodity
	5.13 Slide Rule (Kane *et al.* 2008a)	touch screen access for blind users	ability, accountability, performance, commodity
Web	**5.14 WebAnywhere** (Bigham *et al.* 2008, Bigham *et al.* 2010)	Web access for blind users	ability, accountability, adaptation, transparency, context, commodity

Table 2. Example projects that informed and inspired the formulation of ability-based design and the principles shown above (see Table 1). The projects generally span desktop text entry, mouse pointing, mobile devices, and Web access, and involve users with motor or visual impairments. Ability-based design is not confined to these areas, however, and could be used to address, *e.g.*, cognitive impairments or low literacy.

skills on a physical QWERTY keyboard (Trewin and Pain 1997, Trewin and Pain 1999). In their system, typing data was captured in the background from a user's typing "in the wild" by intercepting key presses before they were sent to the current application. Then, recommendations were made by the system for parameters like key repeat delay, filtering of overlapped keys, and setting key debounce times to prevent accidental double-presses. Trewin and Pain's published system did not directly

configure keyboard parameters, but made suggestions for changing keyboard settings or enabling features such as *Sticky Keys*.

Trewin and Pain's user model contained measures like average and variation of key-press duration, count of double-presses of the same key, counts of workarounds such as using CAPS LOCK for capitalizing only one letter, and so on. Frequencies of English language letter-pairs (digraphs) were included to estimate likelihoods of certain behaviors. An evaluation showed that the model accurately detected long key-press errors, pressing two keys at once, additional key errors, and bounce errors.

Principles. The dynamic keyboard model recommends *adaptations* (3); makes suggestions rather than changing things out-of-sight, giving *transparency* (4); observes and responds to users' *performance* (5); and works with unmodified *commodity* (7) keyboards.

5.2 Invisible Keyguard

An existing "low tech" solution enabling people with motor impairments to type on a standard keyboard is a keyguard, a physical template overlaying a keyboard that prevents users from moving between keys without lifting their hands. Trewin observed that physical keyguards may be stigmatizing and unpopular. Leveraging her dynamic keyboard model, Trewin built an "invisible keyguard" (Trewin 2002), a software system that targeted and prevented overlap errors while avoiding the need for special hardware. Trewin and Pain also introduced *OverlapKeys*, a utility that filtered out additional key-presses automatically, rather than making suggestions for system settings as the dynamic keyboard model did (Trewin and Pain 1997). Three methods for filtering were explored, including the use of heuristics, timing, and language models. Results showed that typing errors could be reduced by 80% using the most successful time-based approach to filtering.

Principles. The Invisible Keyguard and OverlapKeys utility *adapts* (3) to users based on users' *performance* (5). The technologies also work with unmodified *commodity* (7) keyboards.

5.3 Input Device Agent

In creating the *Input Device Agent (IDA)*, Koester *et al.* focused on automatic assessment of and adaptation to users' abilities (Koester *et al.*

2007a). The IDA observed users' typing performance and suggested the best settings for the key repeat rate, repeat delay, and the use of Sticky Keys. The IDA used heuristics to create recommendations for these settings but did not directly change the settings. (A more recent version is the *Keyboard Wizard*, available from Koester Performance Research online.) Results for the IDA showed improvements in text entry error rate and speed.

Koester *et al.* also used the IDA to recommend the optimal mouse gain (Koester *et al.* 2005). The IDA observed lab-based mouse pointing trials and recommended a gain setting accordingly. Results were mixed, but the IDA-recommended gain was helpful for some users and caused no detriment in performance for any users. (A more recent version is called the *Pointing Wizard*, available online.)

An important point about Koester's work was her development of and reliance upon the *Compass* human performance evaluation suite (Koester *et al.* 2003, Koester *et al.* 2006, Koester *et al.* 2007b). In presenting mouse-based performance trials, *Compass* was similar to the test battery used by our SUPPLE automatic interface generator (Gajos *et al.* 2007).
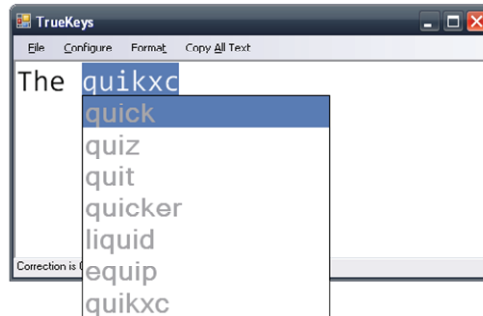
Principles. Both uses of the Input Device Agent provide for *adaptation* (3) in response to measured *performance* (5). In using unmodified keyboards, *commodity* (7) is upheld.

5.4 TrueKeys

An approach to making keyboard typing more accessible, besides using keyboard settings and filters, is to simply correct typing errors as they occur. We built *TrueKeys* (Kane *et al.* 2008b) to do just this. TrueKeys was an adaptive online typing corrector that combined a lexicon of 25,000 English words, string distance algorithms (Levenshtein 1965), and models of keyboard geometry, enabling users with poor dexterity to produce accurate text from inaccurate typing. TrueKeys triggered its correction mechanism after a user finished a word and hit SPACEBAR. TrueKeys also provided an *N*-best list for inspecting and overriding its choices using the arrow keys (Figure 3). In a user study, we found that TrueKeys corrected more errors than the Microsoft Word 2004 spell checker and the open source spell checkers *aspell* and *ispell*.

Principles. TrueKeys supports *transparency* (4) by showing an *N*-best list of possible corrections. It observes user *performance* (5) to automatically correct typing errors. TrueKeys works with unmodified *commodity* (7) keyboards.

Figure 3. TrueKeys' *N*-best list of corrections for the keystrokes "quikxc". Conventional spell checkers would not provide some of the possibilities uncovered by TrueKeys, *e.g.*, "liquid" or "equip," as these suggestions are based on more than lexicographic similarity, including, for example, considerations of keyboard geometry.

5.5 Trackball EdgeWrite

The aforementioned projects made keyboards easier to use, but for many people with spinal cord injuries (SCI), typing on keyboards is simply infeasible. Trackballs are popular among users with SCI, and many users employ an on-screen point-and-click (or point-and-dwell) keyboard to enter text, which is tedious. To address this, we built *Trackball EdgeWrite* (Wobbrock and Myers 2006a, Wobbrock and Myers 2006c, Wobbrock and Myers 2007), which enables users with SCI to write gesturally with their trackballs by "pulsing" the ball in directions corresponding to strokes in the EdgeWrite alphabet (Wobbrock *et al.* 2005), which was originally designed for mobile device users with tremor (Wobbrock *et al.* 2003b). Trackball EdgeWrite maps users' ball motions to vectors corresponding to segments within EdgeWrite letters, and employs adaptive timeouts, slip detection, and word prediction and completion to improve users' accuracy and speed. Its evaluations indicate that Trackball EdgeWrite is faster and more satisfying for text entry than on-screen keyboards.

Principles. Trackball EdgeWrite exhibits *adaptation* (3) to increase accuracy and for word prediction, both of which observe users' *performance* (5). It exposes many preferences supporting *transparency* (4). It uses off-the-shelf trackballs (or any other pointing device), supporting *commodity* (7).
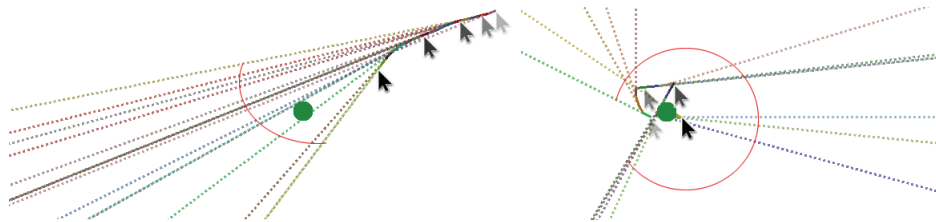
5.6 Steady Clicks

Trewin *et al.*'s *Steady Clicks* utility (Trewin *et al.* 2006) was based on the observation that slips sometimes occur when people with motor impairments press the mouse button (Keates *et al.* 2005). These slips cause the physical mouse to move, which, in turn, causes the on-screen cursor to move out of its intended target. Steady Clicks prevented slips by freezing the mouse cursor in place when the button went down. It also suppressed accidental clicks that occurred while the mouse was in motion or from secondary mouse buttons. These simple adjustments resulted in significant speed and accuracy gains for participants with motor impairments. Since then, Steady Clicks has been explored for use with pen-based computers such as Tablet PCs, with positive results for older users (Moffatt and McGrenere 2010).

Principles. Steady Clicks is based on and responds to users' *performance* (5). It works to improve an unmodified *commodity* (7) mouse or trackball.

5.7 Angle Mouse

Like with Steady Clicks (Trewin *et al.* 2006), we took up the challenge of improving mouse, touchpad, and trackball performance through software alone. In our own studies (Wobbrock and Gajos 2007) and those of others (Hwang *et al.* 2004), we saw that for people with motor impairments, the ballistic phase of pointing, in which the user initially "shoots" the mouse cursor toward a target, was reasonably accurate, but the corrective phase, in which the user makes fine adjustments, was often inaccurate due to high neuromotor noise (Harris and Wolpert 1998, Walker *et al.* 1997). In particular, during corrective-phase pointing, highly divergent angles were created by the cursor's movement trajectory. To capitalize on this information, we created the *Angle Mouse* (Wobbrock *et al.* 2009), which observed the angles formed by the mouse during movement and adjusted the mouse control-display gain in response. When the spread of angles, or *angular deviation*, was low (Figure 4, left), as in ballistic-phase pointing, the gain was kept high. When the spread of angles was high (Figure 4, right), as in corrective-phase pointing, the gain was dropped, which increased the target size in motor-space. Importantly,

these adjustments were made without any knowledge of the targets, making the Angle Mouse a *target-agnostic* pointing facilitation technique capable of being deployed on real-world systems without support from operating systems or applications.

Figure 4. Visualizations showing how the Angle Mouse works. In the left image, the cursor path is relatively coherent, and the angular deviation, marked by the red arc, is relatively low. The gain is therefore kept high. In the right image, the cursor path is divergent due to numerous path corrections, creating high angular deviation. The gain is dropped to effectively magnify the target in motor-space.
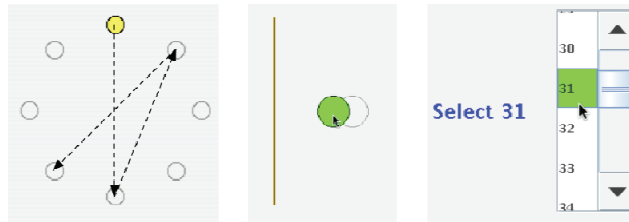
Results for participants with motor impairments showed that using the Angle Mouse, throughput, a combined speed-accuracy measure (MacKenzie and Isokoski 2008), was more than 10% higher than the Windows default mouse. The Angle Mouse was also about 18% more accurate and made significantly fewer final-stage corrective submovements than the Windows default mouse. Results also showed that the performance of nondisabled users was not significantly affected by the Angle Mouse because nondisabled users did not exhibit the motor-control difficulties that caused the Angle Mouse to adjust the mouse gain.

Principles. The Angle Mouse *adapts* (3) the control-display gain based on continually monitored user *performance* (5). It works with unmodified *commodity* (7) pointing devices. A downloadable version offers an interface for inspecting parameters, providing *transparency* (4).
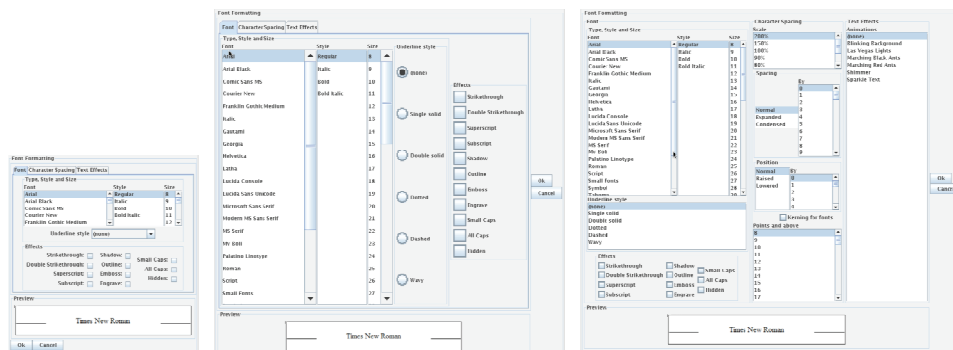
5.8 SUPPLE

Perhaps our deepest example of ability-based design is SUPPLE (Gajos and Weld 2004, Gajos and Weld 2005, Gajos and Weld 2006, Gajos *et al.* 2007, Gajos *et al.* 2008, Gajos *et al.* 2010). Unlike Steady Clicks and the Angle Mouse, which attempted to improve mouse pointing, SUPPLE opted to make the interface on which the mouse acts more accessible. SUPPLE automatically generated user interfaces using an optimization approach that intelligently searched the space of possible interfaces for the one that minimized the movement time of the user. SUPPLE constructed a model of a user's movement performance from a short one-time battery of clicking,

pointing, dragging, and list selection tasks (Figure 5). Because Fitts' law (Fitts 1954) did not always model our participants with motor impairments well, SUPPLE automatically selected features for custom regression models for each user. These custom regression models more accurately predicted the performance of users with motor impairments and people using unusual input devices such as eye-trackers or head mice.

Figure 5. SUPPLE's pointing, dragging, and list selection tasks from the ability-assessment battery. From these tasks, SUPPLE builds a predictive model of a user's movement time that is used to parameterize a search for the "optimal" interface. Widget selection, size, position, and grouping are all part of the search space.

The user interfaces generated by SUPPLE were automatically created in a matter of seconds without using any heuristic knowledge of health conditions or typical functional profiles. Only the results of the user's short performance test were used by the generator. Nonetheless, SUPPLE interfaces clearly reflected the functional differences among participants (Figure 6). SUPPLE also gave users the ability to override and customize design choices made by the system. Our study results showed that with SUPPLE-generated interfaces, motor-impaired users were 28% faster, 73% more accurate, and significantly more satisfied than with manufacturers' defaults (Gajos *et al.* 2008).

Figure 6. The left interface is the default font-format dialog box from Microsoft Word, rebuilt in SUPPLE. The same functionality is provided in the middle interface, which was automatically generated for a person with spastic Cerebral Palsy. The right interface was generated for a person with Muscular Dystrophy. The middle interface contains big targets and longer movement distances, and uses multiple tab panes to group features. The

right interface contains small targets at close distances and no tab panes. SUPPLE makes these choices based only on user performance without any heuristic knowledge of Cerebral Palsy or Muscular Dystrophy.

Principles. SUPPLE *adapts* (3) interfaces to users based on their *performance* (5) in a test battery. It gives users the power to inspect and override its choices, resulting in good *transparency* (4). SUPPLE allows *commodity* (7) pointing devices to perform faster and more accurately.

5.9 Automatic Mouse Pointing Assessment

Whereas SUPPLE (Gajos *et al.* 2010) required users to perform pointing tasks in a controlled battery, Hurst *et al.* sought to measure and model pointing performance in naturalistic tasks. One such project measured menu performance with the mouse and, without any underlying task knowledge, was able to discern novice from skilled computer users with 91% accuracy (Hurst *et al.* 2007). The authors built a menu-driven drawing application in which help information changed based on the inferred expertise of the user, with good results. A follow-up project automatically distinguished between users with and without pointing problems, young and old users and users with Parkinson's disease, and users needing and not needing specific adaptations, all with accuracies above 90% (Hurst *et al.* 2008a). In both projects, Hurst *et al.* used wrapper-based feature selection and a C4.5 decision tree to create parsimonious learned statistical models of user performance.

Principles. Hurst *et al.*'s menu-driven drawing application upholds *adaptation* (3), while both projects overall strongly uphold *performance* (5). In seeking to make regular mice more effective, both projects also uphold *commodity* (7).

5.10 VoiceDraw

Many of the aforementioned projects attempted to improve mouse pointing by adjusting either the mouse or the interface beneath it. Instead, *VoiceDraw* enabled the mouse cursor to be controlled by the human voice (Harada *et al.* 2007). Drawing and painting with conventional speech recognition is poor because of the need to map *discrete* voice commands to *continuous* paintbrush movements. VoiceDraw utilized continuous non-speech vowel sounds to fluidly move a paintbrush in a custom drawing application. A vowel-sound mapping was used by the underlying *Vocal*

Joystick (Harada *et al.* 2006), providing a "vowel map" (Figure 7) in which each sound could be morphed into the sounds around it, making it possible for a user to smoothly move the paintbrush in any direction.

   Principles. VoiceDraw *adapts* (3) itself to the vowel productions of the user, which is part of the user's *performance* (5). In using a regular computer microphone, VoiceDraw also embodies *commodity* (7).
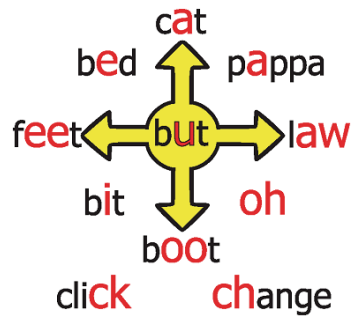

Figure 7. The vowel map used in VoiceDraw allowing for continuous movement control.

5.11 Barrier Pointing

   Mobile interfaces are full of tiny targets, which may pose accessibility challenges. Whereas the customary approach of "flying in" to tap a small target with a stylus or finger is difficult for many people with motor impairments, utilizing physical screen edges provides an opportunity for increasing stability during pointing. Indeed, virtual impenetrable screen edges have shown target acquisition benefits in desktop user interfaces (Appert *et al.* 2008, Farris *et al.* 2001, Farris *et al.* 2002a, Farris *et al.* 2002b, Johnson *et al.* 2003, Walker and Smelcer 1990). Our goal was to capitalize on the physical screen bezels of existing mobile devices (Wobbrock 2003, Wobbrock *et al.* 2003a). Rather than requiring device add-ons, we devised a set of techniques we called *barrier pointing* (Froehlich *et al.* 2007).

   Our barrier pointing techniques relied on the ability for a user to press a stylus firmly against an elevated screen bezel to provide stability during motion. *Barrier targets* (or *barrier widgets*) were placed along the screen bezel and could be actuated according to the particular barrier pointing method used. In our most successful method, the stylus (or finger) could slide along the edge and *lift* to acquire the target. The approach of lifting

to select has been called *lift-off* or *take-off selection* (Potter *et al.* 1988, Sears and Shneiderman 1991, Vogel and Baudisch 2007), although it has not been utilized along screen edges for improving accessibility. In a case study, one participant with spastic Cerebral Palsy had an error rate of 66.7% using the conventional "fly-in and tap" technique, but only 13.3% using lift-off selection with barrier targets. Also, he was 48.5% faster with lift-off selection along the physical edges than with "fly-in and tap."

Figure 8 shows a music player concept design that uses barrier pointing for improving accessibility. Such a design may also help people who are not physically impaired, but situationally impaired (*e.g.*, by walking; see section 5.12.)

Principles. The development of barrier pointing was based on the careful measurement of users' *performance* (5). In using off-the-shelf mobile devices, it upholds *commodity* (7).
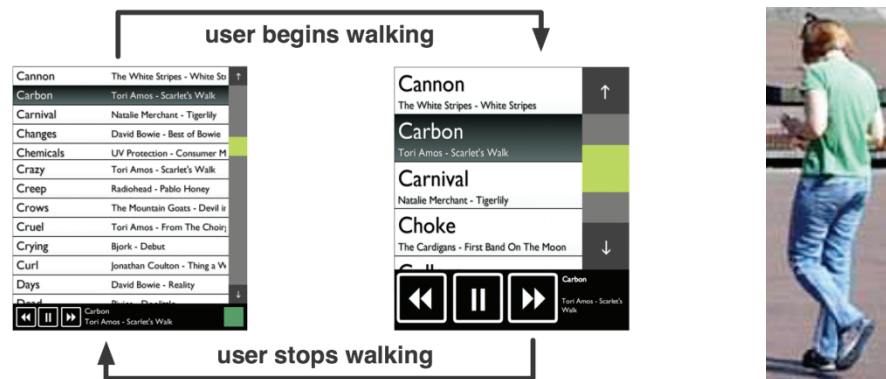

Figure 8. The left design is a typical music player, with targets near the screen bottom in "open space." The right design has targets along the left and right screen edges, where elevated physical bezels improve accuracy when used in conjunction with the *lift-off* selection technique. Although round, the right-side targets would be made responsive all the way to the screen edge.

5.12 Walking User Interfaces

A person's ability is not determined solely by his or her health, but also by the current environment or context (Newell 1995, Sears *et al.* 2003). Prior research has shown that using a mobile device while walking can impair a user's ability to read text (Mustonen *et al.* 2004) or select targets (Lin *et al.* 2007). An ability-based design perspective suggests that we adapt mobile user interfaces to match a user's "situationally-impaired" abilities while walking (Yamabe and Takahashi 2007). We explored *walking user interfaces* (Kane *et al.* 2008c), which detected whether a user

was currently walking, and automatically adjusted the interface layout to compensate (Figure 9). When standing, the user was given relatively small text and targets, allowing large amounts of content to be displayed on the screen. When walking, the user was given larger text and controls, but fewer items on the screen, requiring more scrolling. Our study results were mixed; walking user interfaces that adapted target size and level of detail showed signs of improving performance while moving, but the improvements in some cases were minimal, and further exploration of this design space is needed.

Principles. Walking user interfaces *adapt* (3) to their users based on changing *context* (6). They uphold *commodity* (7) by functioning on unmodified mobile devices.

Figure 9. Our walking user interface adapts to a user's situationally-impaired visual and motor abilities. When the user begins walking, the on-screen text and targets increase in size. On the right, a participant in our user study is shown walking an outdoor course with our mobile device. Although our prototype detected standing *vs.* walking using the device's camera, for our study we simulated adaptation using the Wizard-of-Oz technique.

5.13 Slide Rule

Today, touch screens appear on grocery-store and airport kiosks, microwave ovens, dishwasher panels, new bank ATMs, and of course, mobile devices. How can a blind person use a touch screen? This was the question motivating our creation of *Slide Rule*, a prototype utilizing accuracy-relaxed multi-touch gestures, "finger reading," and screen layout schemes to enable blind people to use unmodified touch screens (Kane *et al.* 2008a).

Slide Rule offered three applications: *Phonebook*, *Mail*, and *Music*. We prototyped these on an Apple iPod Touch. In contrast to a key- or button-

based screen reader, audio output was controlled by moving a "reading finger" across the screen. The spoken audio read the screen at a level of detail proportional to the speed of the moving finger. For example, if the finger moved quickly down an alphabetical list of Phonebook contacts, the spoken audio would say only the first letter of the names: "A," "E," "G", "L" and so on. If the finger moved more slowly, last names would be read. If it moved even more slowly, both last and first names would be read. For our prototyping purposes, no visual interfaces were drawn to the screen, although for sighted users, such an option easily could be added.

We found that when a blind user attempted to lift-and-tap a target on a touch screen, the user lost a sense of location. To address this, Slide Rule used a method of target selection we called *second-finger-tap*: while the reading finger paused over the desired target, another finger (or thumb) on the same (or opposite) hand could tap *anywhere* on the screen to trigger the item beneath the reading finger. This approach lessened the accuracy demands for acquiring targets and allowed the user to interact with the interface without losing a sense of location. Second-finger-tap was incorporated into Apple's *VoiceOver* software as *split-tap* and in subsequent research projects like *No-Look Notes* (Bonner *et al.* 2010).

Principles. Slide Rule has as its basis for design the *performance* (5) of blind users. Slide Rule uses *commodity* (7) iPod Touch hardware. Incidentally, Slide Rule is one of our best examples of *ability* (1) in leveraging multi-touch gestures for blind users who are accustomed to using their fingers to navigate and negotiate the world.

5.14 WebAnywhere

WebAnywhere was a Web site that reads aloud other Web sites, serving as a Web-based screen reader for the Web itself (Bigham and Prince 2007, Bigham *et al.* 2008). A significant advantage of WebAnywhere over existing screen readers was that WebAnywhere requires no additional software to be installed, allowing it to be used on shared terminals in libraries, classrooms, agencies, and so on. From the start, WebAnywhere enabled users to store and retrieve personal preference profiles, including customizations of WebAnywhere's keyboard shortcuts and reading behavior. Recent versions of WebAnywhere added support for multiple languages and speech recognition (Bigham *et al.* 2010). Blind users played

a formative role in the design of WebAnywhere, giving suggestions for its many features. Today, WebAnywhere enjoys a sizeable open source developer community.

Principles. In its extensive support for users' personal interface preferences, including language preferences, WebAnywhere upholds both *adaptation* (3) and *transparency* (4). By proactively inspecting users' systems and adjusting its function to suit (*e.g.*, working with available sound players), WebAnywhere upholds *context* (6). WebAnywhere strongly upholds *commodity* (7) by using an unmodified Web browser without any software installation required.

6. DISCUSSION

The previous section reviewed projects that informed and inspired ability-based design and its principles. These projects also, in various ways and to various degrees, upheld the principles of ability-based design. All 14 projects upheld *ability* (1), *accountability* (2), and *commodity* (7); 12 upheld *performance* (5), and 10 upheld *adaptation* (3); *transparency* (4) and *context* (6) were upheld by 6 and 2 projects, respectively. All projects were grounded in firm notions of *what users can do*, capitalizing on the abilities of users to act and making the ability-based designs accommodate that action.

Many of our projects required us to think about everyday devices anew. With VoiceDraw (section 5.10), we had to utilize a "speech recognizer" for *non*-speech sounds. With barrier pointing (section 5.11), we had to see the elevated physical bezels around the screen as holding new opportunities for design.

The reviewed projects exhibited a range of adaptivity by the system and adaptability by the user. *Low adaptivity* projects made immediate, local changes. One example was the adaptive gain setting in the Angle Mouse (Wobbrock *et al.* 2009), which was based on the most recent mouse movement trajectory. *High adaptivity* projects made lasting, global adaptations, often based on a performance model. SUPPLE was an example of a system with high adaptivity (Gajos *et al.* 2010). Still other projects were user-adaptable, having customizable settings that affected the software's performance. WebAnywhere was a good example of highly

customizable software that could be tailored to the skills and preferences of its users (Bigham *et al.* 2008).

Eleven of our 14 examples focused on users with motor impairments, but such a focus is not required for ability-based design. All users have abilities, and to the extent that context or impairment affect those abilities, ability-based design should be relevant. For example, one could create a SUPPLE-like test battery (Gajos *et al.* 2007) for low-vision or hard-of-hearing users to enable a computer to create a model of a user's visual or auditory skills and adapt interfaces accordingly.

It was a common experience for us while developing some of the aforementioned technologies to encounter skepticism and even resistance on the part of participants who were dubious that unmodified input devices could be used by them. Perhaps the participants had grown accustomed to thinking that they would always have trouble with devices meant "for everyone else." Whatever the case, after considerable iterative design with heavy input from participants, we saw genuine delight by many of our users when they could use "normal" devices with new software tailored to their abilities.

7. RESEARCH CHALLENGES

This article attempts to establish the nature and purpose of ability-based design, and put forth its guiding principles. We believe that as a concept, ability-based design may go far beyond any of the reviewed projects to realize a day when all software, and perhaps even hardware, will be perfectly tailored to its user and his or her abilities. Strikingly, this is quite the opposite of universal design; rather, it is the *universal application of "design-for-one"* (Harper 2007).

So what is required to make the "universal application of 'design-for-one' " a reality? Of central importance is work on automatically detecting, assessing, and understanding people's abilities (Casali 1995, Gajos *et al.* 2007, Gajos *et al.* 2008, Hurst *et al.* 2008a, Keates *et al.* 2002, Law *et al.* 2005, Price and Sears 2008). Of special importance is the need for quick, low-effort, accurate, and reliable performance tests that can be administered once for most users, or periodically for people with conditions that change over time. A more challenging but more useful solution would be to accurately measure users' abilities from performance

"in the wild;" that is, outside an artificial test battery (Chapuis *et al.* 2007, Hurst *et al.* 2008b). However, accurately measuring tasks in the wild requires inferring the *intention* of tasks, requiring, among other things, the ability to know what a user is looking at, trying to point to, attempting to write, and so on. Algorithms for segmenting real-world mouse movement into discrete aimed pointing movements are necessary, but they are only a start. We also need to know where targets in the interface lie (Dixon and Fogarty 2010, Hurst *et al.* 2010), and what mouse behavior constitutes an error. The same is true for understanding text entry without the benefits of a laboratory study where phrases are presented to participants for transcription; prior work has made some progress here (Wobbrock and Myers 2006b).

Related to ability measurement is the challenge of sensing context. Future work, especially on mobile accessibility, should consider how to make devices more aware of and responsive to environmental factors like lighting, temperature, and ambient noise (Wobbrock 2006). While "on the go," mobile device users may experience multiple modes of transportation, from walking to riding a bus to driving a car. They may be stationary or moving up or down stairs. They may be outdoors, exposed to cold temperatures (and the need for gloves), rain water, glare, wind, ambient noise, and so on. Mobile device users' social contexts also change, from being in a movie theater to business meetings to personal conversations. Current mobile devices are mostly ignorant of these factors and require users to adapt their behavior to the environment. Recent advances in mobile activity recognition hold promise for enabling devices to become more aware of their users' contexts (Choudhury *et al.* 2008, Hinckley *et al.* 2000).

Once performance has been accurately measured or context has been sensed, there still remains the considerable challenge of modeling a user's abilities. For systems to create, modify, and assess a predictive model of a user's abilities is still an ongoing challenge (Biswas and Robinson 2008), especially for users with impairments, whose abilities are greatly varied, even for the same medical condition. For this reason, conventional user models do not seem to hold well for many people with impairments (Keates *et al.* 2002). Because variability among such users is high,

techniques for accommodating this variability in user models are necessary for advancing ability-based design. Work on perceptive user interfaces (Hwang *et al.* 2001) is still in its early stages, but could help systems understand the state, context, and abilities of users. Recent advances in automatically detecting the impairments of users also hold promise (Hurst *et al.* 2008a).

Assuming we can accurately measure and model users' abilities and context, the next question is one of design: how best can we employ this knowledge? More research is needed in user interface adaptation. Although SUPPLE makes important advances (Gajos *et al.* 2010), it leaves much unanswered, like how to incorporate errors, visual search time, input device characteristics, and aesthetic concerns.

Another design approach is to allow end-user adaptation in a more flexible manner. Although most applications contain some configuration options, applications must offer a much wider range of possibilities to fully support ability-based design. Of particular importance is providing feedback in the form of *previews* so that users know the results of their changes (Terry and Mynatt 2002). Showing previews can be difficult for changes that alter complex motor-oriented parameters or other non-visual interface aspects. However, previews could improve the trial-and-error process users currently endure when adapting software.

Finally, we need to further investigate how commodity input devices can be repurposed in novel ways for people with disabilities. Hardware researchers need to devise more flexible input devices that can be used in various ways. Reconfigurable devices may hold promise for greater adaptability (Schwesig *et al.* 2004), although they have yet to be realized. Software that can remap device inputs to required outputs may also make devices more versatile (Carter *et al.* 2006, Wang and Mankoff 2003).

In the end, the progress necessary for ability-based design to flourish will leave few areas of computing untouched. It is our hope that ability-based design can serve as a unifying grand challenge for fruitful collaborations in computing, human factors, psychology, design, and human-computer interaction.

8. CONCLUSION

This article presented ability-based design as a refinement to, and

refocusing of, prior approaches to accessible design. Just as user-centered design shifted the focus of interactive system design from systems to users, ability-based design shifts the focus of accessible design from disability to ability. This article put forth seven ability-based design principles and described fourteen projects that informed and inspired these principles. Future research directions that touch many areas of computing were also provided.

It is our hope that this work will serve as a vantage point from which to envision a future consisting of tools, techniques, methods, and models that better leverage the malleability of software, the availability of hardware, and the abilities and potential of all human users.

9. ACKNOWLEDGEMENTS

REFERENCES

Appert, C., Chapuis, O. and Beaudouin-Lafon, M. (2008). Evaluation of pointing performance on screen edges. *Proceedings of the ACM Conference on Advanced Visual Interfaces (AVI '08)*. New York: ACM Press, pp. 119-126.

Bates, R. and Istance, H. O. (2003). Why are eye mice unpopular? A detailed comparison of head and eye controlled assistive technology pointing devices. *Universal Access in the Information Society 2* (3), pp. 280-290.

Bigham, J. P. and Prince, C. M. (2007). WebAnywhere: A screen reader on-the-go. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '07)*. New York: ACM Press, pp. 225-226.

Bigham, J. P., Prince, C. M. and Ladner, R. E. (2008). WebAnywhere: A screen reader on-the-go. *Proceedings of the ACM Int'l Cross-Disciplinary Conference on Web Accessibility (W4A '08)*. New York: ACM Press, pp. 73-82.

Bigham, J. P., Chisholm, W. and Ladner, R. E. (2010). WebAnywhere: Experiences with a new delivery model for access technology. *Proceedings of the ACM Int'l Cross-Disciplinary Conference on Web Accessibility (W4A '10)*. New York: ACM Press. No. 15.

Biswas, P. and Robinson, P. (2008). Automatic evaluation of assistive interfaces. *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '08)*. New York: ACM Press, pp. 247-256.

Bonner, M. N., Brudvik, J. T., Abowd, G. D. and Edwards, W. K. (2010). No-Look Notes: Accessible eyes-free multi-touch text entry. *Proceedings of the 8th Int'l Conference on Pervasive Computing (Pervasive '10)*. Heidelberg: Springer-Verlag, pp. 409-426.

Carter, S., Hurst, A., Mankoff, J. and Li, J. (2006). Dynamically adapting GUIs to diverse input devices. *Proceedings of the ACM SIGACESS Conference on Computers and Accessibility (ASSETS '06)*. New York: ACM Press, pp. 63-70.

Casali, S. P. (1992). Cursor control device performance by persons with physical disabilities: Implications for hardware and software design. *Proceedings of the 36th Annual Meeting of the Human Factors Society*. Santa Monica, California: Human Factors Society, pp. 311-315.

Casali, S. P. (1995). A physical skills based strategy for choosing an appropriate interface method. In *Extra-Ordinary Human-Computer Interaction: Interfaces for Users with Disabilities*, A. D. N. Edwards (ed). Cambridge, England: Cambridge University Press, pp. 315-341.

Chapuis, O., Blanch, R. and Beaudouin-Lafon, M. (2007). Fitts' law in the wild: A field study of aimed movements. *LRI Technical Repport Number 1480* (December 2007). Orsay, France: Universite de Paris Sud.

Chickowski, E. (2004). It's all about access. *Alaska Airlines Magazine 28* (12), pp. 26-31, 80-82.

Choudhury, T., Borriello, G., Consolvo, S., Haehnel, D., Harrison, B., Hemingway, B., Hightower, J., Klasnja, P., Koscher, K., LaMarca, A., Landay, J. A., LeGrand, L., Lester, J., Rahimi, A., Rea, A. and Wyatt, D. (2008). The Mobile Sensing Platform: An embedded activity recognition system. *IEEE Pervasive Computing 7* (2), pp. 32-41.

Cook, A. M. and Hussey, S. M. (2002). *Assistive Technologies: Principles and Practice* (2nd ed.). St. Louis: Mosby.

Dawe, M. (2004). Complexity, cost and customization: Uncovering barriers to adoption of assistive technology. *Refereed Poster at the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '04)*. Atlanta, Georgia (October 18-20, 2004).

Dixon, M. and Fogarty, J. A. (2010). Prefab: Implementing advanced behaviors using pixel-based reverse engineering of interface structure. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '10)*. New York: ACM Press, pp. 1525-1534.

Edwards, A. D. N. (1995). Computers and people with disabilities. In *Extra-Ordinary Human-Computer Interaction: Interfaces for Users with Disabilities*, A. D. N. Edwards (ed). Cambridge, England: Cambridge University Press, pp. 19-43.

Evenson, S., Rheinfrank, J. and Dubberly, H. (2010). Ability-centered design: From static to adaptive worlds. *interactions 17* (6), pp. 75-79.

Farris, J. S., Jones, K. S. and Anders, B. A. (2001). Acquisition speed with targets on the edge of the screen: An application of Fitts' Law to commonly used web browser controls. *Proceedings of the Human Factors and Ergonomics Society 45th Annual Meeting (HFES '01)*. Santa Monica, California: Human Factors and Ergonomics Society, pp. 1205-1209.

Farris, J. S., Jones, K. S. and Anders, B. A. (2002a). Using impenetrable borders in a graphical web browser: How does distance influence target selection speed? *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting (HFES '02)*. Santa Monica, California: Human Factors and Ergonomics Society, pp. 1300-1304.

Farris, J. S., Jones, K. S. and Anders, B. A. (2002b). Using impenetrable borders in a graphical web browser: Are all angles equal? *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting (HFES '02)*. Santa Monica, California: Human Factors and Ergonomics Society, pp. 1251-1255.

Fichten, C. S., Barile, M., Asuncion, J. V. and Fossey, M. E. (2000). What government, agencies, and organizations can do to improve access to computers for postsecondary students with disabilities: Recommendations based on Canadian empirical data. *International Journal of Rehabilitation Research 23* (3), pp. 191-199.

Findlater, L. and McGrenere, J. (2004). A comparison of static, adaptive, and adaptable menus. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '04)*. New York: ACM Press, pp. 89-96.

Findlater, L., Moffatt, K., McGrenere, J. and Dawson, J. (2009). Ephemeral adaptation: The use of gradual onset to improve menu selection performance. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '09)*. New York: ACM Press, pp. 1655-1664.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology 47* (6), pp. 381-391.

Friedman, B., Kahn, P. H. and Borning, A. (2006). Value Sensitive Design and Information Systems. In *Human-Computer Interaction in Management Information Systems: Foundations*, D. Galletta and P. Zhang (eds). Armonk, New York: M.E. Sharpe, pp. 348-372.

Froehlich, J., Wobbrock, J. O. and Kane, S. K. (2007). Barrier Pointing: Using physical edges to assist target acquisition on mobile device touch screens. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '07)*. New York: ACM Press, pp. 19-26.

Gajos, K. and Weld, D. S. (2004). SUPPLE: Automatically generating user interfaces. *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '04)*. New York: ACM Press, pp. 93-100.

Gajos, K. and Weld, D. S. (2005). Preference elicitation for interface optimization. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '05)*. New York: ACM Press, pp. 173-182.

Gajos, K. and Weld, D. S. (2006). Automatically generating custom user interfaces for users with physical disabilities. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '06)*. New York: ACM Press, pp. 243-244.

Gajos, K. Z., Wobbrock, J. O. and Weld, D. S. (2007). Automatically generating user interfaces adapted to users' motor and vision capabilities. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '07)*. New York: ACM Press, pp. 231-240.

Gajos, K. Z., Wobbrock, J. O. and Weld, D. S. (2008). Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '08)*. New York: ACM Press, pp. 1257-1266.

Gajos, K. Z., Weld, D. S. and Wobbrock, J. O. (2010). Automatically generating personalized user interfaces with SUPPLE. *Artificial Intelligence 174* (12-13), pp. 910-950.

Goette, T. (1998). Factors leading to the successful use of voice recognition technology. *Proceedings of the ACM SIGCAPH Conference on Assistive Technologies (ASSETS '98)*. New York: ACM Press, pp. 189-196.

Gould, J. D. and Lewis, C. (1985). Designing for usability: Key principles and what designers think. *Communications of the ACM 28* (3), pp. 300-311.

Gross, D. P. (2004). Measurement properties of performance-based assessment of functional capacity. *Journal of Occupational Rehabilitation 14* (3), pp. 165-174.

Harada, S., Landay, J. A., Malkin, J., Li, X. and Bilmes, J. A. (2006). The Vocal Joystick: Evaluation of voice-based cursor control techniques. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '06)*. New York: ACM Press, pp. 197-204.

Harada, S., Wobbrock, J. O. and Landay, J. A. (2007). VoiceDraw: A hands-free voice-driven drawing application for people with motor impairments. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '07)*. New York: ACM Press, pp. 27-34.

Harper, S. (2007). Is there design for all? *Universal Access in the Information Society 6* (1), pp. 111-113.

Harris, C. M. and Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature 394*, pp. 780-784.

Hazard, B. L. (2008). Separate but equal? A comparison of content on library web pages and their text versions. *Journal of Web Librarianship 2* (2-3), pp. 417-428.

Hinckley, K., Pierce, J., Sinclair, M. and Horvitz, E. (2000). Sensing techniques for mobile interaction. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '00)*. New York: ACM Press, pp. 91-100.

Hurst, A., Hudson, S. E. and Mankoff, J. (2007). Dynamic detection of novice vs. skilled use without a task model. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '07)*. New York: ACM Press, pp. 271-280.

Hurst, A., Hudson, S. E., Mankoff, J. and Trewin, S. (2008a). Automatically detecting pointing performance. *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '08)*. New York: ACM Press, pp. 11-19.

Hurst, A., Mankoff, J. and Hudson, S. E. (2008b). Understanding pointing problems in real world computing environments. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '08)*. New York: ACM Press, pp. 43-50.

Hurst, A., Hudson, S. E. and Mankoff, J. (2010). Automatically identifying targets users interact with during real world tasks. *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '10)*. New York: ACM Press, pp. 11-20.

Hwang, F., Keates, S., Langdon, P., Clarkson, P. J. and Robinson, P. (2001). Perception and haptics: Towards more accessible computers for motion-impaired users. *Proceedings of the ACM Workshop on Percetive User Interfaces (PUI '01)*. New York: ACM Press, pp. 1-9.

Hwang, F., Keates, S., Langdon, P. and Clarkson, P. J. (2004). Mouse movements of motion-impaired users: A submovement analysis. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '04)*. New York: ACM Press, pp. 102-109.

Jipp, M., Badreddin, E., Abkai, C. and Hesser, J. (2008a). Individual ability-based system configuration: Cognitive profiling with Bayesian networks. *Proceedings of the IEEE Int'l Conference on Systems, Man and Cybernetics (SMC '08)*. Piscataway, New Jersey: IEEE Press, pp. 3359-3364.

Jipp, M., Wagner, A. and Badreddin, E. (2008b). Individual ability-based system design of dependable human-technology interaction. *Proceedings of the 17th IFAC World Congress (IFAC '08)*. Oxford, England: Elsevier Ltd., pp. 14779-14784.

Jipp, M., Bartolein, C. and Badreddin, E. (2009a). Predictive validity of wheelchair driving behavior for fine motor abilities: Definition of input variables for an adaptive wheelchair system. *Proceedings of the IEEE Int'l Conference on Systems, Man and Cybernetics (SMC '09)*. Piscataway, New Jersey: IEEE Press, pp. 39-44.

Jipp, M., Bartolein, C., Badreddin, E., Abkai, C. and Hesser, J. (2009b). Psychomotor profiling with Bayesian neworks: Prediction of user abilities based on inputs of motorized wheelchair parameters. *Proceedings of the IEEE Int'l Conference on Systems, Man and Cybernetics (SMC '09)*. Piscataway, New Jersey: IEEE Press, pp. 1680-1685.

Johnson, B. R., Farris, J. S. and Jones, K. S. (2003). Selection of web browser controls with and without impenetrable borders: Does width make a difference? *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting (HFES '03)*. Santa Monica, California: Human Factors and Ergonomics Society, pp. 1380-1384.

Kane, S. K., Bigham, J. P. and Wobbrock, J. O. (2008a). Slide Rule: Making mobile touch screens accessible to blind people using multi-touch interaction techniques. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '08)*. New York: ACM Press, pp. 73-80.

Kane, S. K., Wobbrock, J. O., Harniss, M. and Johnson, K. L. (2008b). TrueKeys: Identifying and correcting typing errors for people with motor impairments. *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '08)*. New York: ACM Press, pp. 349-352.

Kane, S. K., Wobbrock, J. O. and Smith, I. E. (2008c). Getting off the treadmill: Evaluating walking user interfaces for mobile devices in public spaces. *Proceedings of the ACM Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '08)*. New York: ACM Press, pp. 109-118.

Keates, S., Clarkson, P. J., Harrison, L.-A. and Robinson, P. (2000). Towards a practical inclusive design approach. *Proceedings of the ACM Conference on Universal Usability (CUU '00)*. New York: ACM Press, pp. 45-52.

Keates, S., Langdon, P., Clarkson, P. J. and Robinson, P. (2002). User models and user physical capability. *User Modeling and User-Adapted Interaction 12* (2-3), pp. 139-169.

Keates, S. and Clarkson, P. J. (2003). Countering design exclusion through inclusive design. *Proceedings of the ACM Conference on Universal Usability (CUU '03)*. New York: ACM Press, pp. 69-76.

Keates, S. and Clarkson, J. (2004). *Countering Design Exclusion: An Introduction to Inclusive Design*. London, UK: Springer-Verlag.

Keates, S., Trewin, S. and Paradise, J. (2005). Using pointing devices: Quantifying differences across user groups. *Proceedings of the 11th International Conference on Human-Computer Interaction (HCI Int'l '05)*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Kelley, D. and Hartfield, B. (1996). The designer's stance. In *Bringing Design to Software*, T. Winograd (ed). Reading, MA: Addison-Wesley, pp. 151-170.

Koester, H. H. (2003). Abandonment of speech recognition by new users. *Proceedings of the RESNA 26th Annual Conference (RESNA '03)*. Arlington, Virginia: RESNA Press.

Koester, H. H., LoPresti, E. F., Ashlock, G., McMillan, W. W., Moore, P. and Simpson, R. C. (2003). Compass: Software for computer skills assessment. *Proceedings of CSUN's 18th Annual Conference on Technology and Persons with Disabilities (CSUN '03)*. California State University Northridge.

Koester, H. H., LoPresti, E. and Simpson, R. C. (2005). Toward Goldilocks' pointing device: Determining a "just right" gain setting for users with physical impairments. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '05)*. New York: ACM Press, pp. 84-89.

Koester, H. H., LoPresti, E. F. and Simpson, R. C. (2006). Measurement validity for Compass assessment software. *Proceedings of the RESNA 29th Annual Conference (RESNA '06)*. Arlington, Virginia: RESNA Press.

Koester, H. H., LoPresti, E. and Simpson, R. C. (2007a). Toward automatic adjustment of keyboard settings for people with physical impairments. *Disability and Rehabilitation: Assistive Technology 2* (5), pp. 261-274.

Koester, H. H., Simpson, R. C., Spaeth, D. M. and LoPresti, E. F. (2007b). Reliability and validity of Compass software for access assessment. *Proceedings of the RESNA 30th Annual Conference (RESNA '07)*. Arlington, Virginia: RESNA Press.

Kondraske, G. V. (1988a). Workplace design: An elemental resource approach to task analysis and human performance measurements. *Proceedings of the Int'l Conference of the Association for the Advancement of Rehabilitation Technology (ICAART '88)*. Washington, D.C.: RESNA Press, pp. 608-611.

Kondraske, G. V. (1988b). Rehabilitation engineering: Towards a systematic process. *IEEE Engineering in Medicine and Biology Magazine 7* (3), pp. 11-15.

Kondraske, G. V. (1990a). A PC-based performance measurement laboratory system. *Journal of Clinical Engineering 15* (6), pp. 467-478.

Kondraske, G. V. (1990b). Quantitative measurement and assessment of performance. In *Rehabilitation Engineering*, J. H. Leslie and R. V. Smith (eds). Boca Raton, Florida: CRC Press, pp. 101-125.

Kondraske, G. V. (1995). A working model for human system-task interfaces. In *The Biomedical Engineering Handbook*, J. D. Bronzino (ed). Boca Raton, Florida: CRC Press, pp. 2157-2174.

Law, C. M., Sears, A. and Price, K. J. (2005). Issues in the categorization of disabilities for user testing. *Proceedings of the 11th International Conference on Human-Computer Interaction (HCI Int'l '05)*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Lazar, J. (2007). *Universal Usability: Designing Computer Interfaces for Diverse Users*. West Sussex, England: John Wiley & Sons.

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR 163* (4), pp. 845-848.

Lin, M., Goldman, R., Price, K. J., Sears, A. and Jacko, J. (2007). How do people tap when walking? An empirical investigation of nomadic data entry. *International Journal of Human-Computer Studies 65* (9), pp. 759-769.

Mace, R. L., Hardie, G. J. and Place, J. P. (1991). Accesible environments: Toward universal design. In *Design Intervention: Toward a More Humane Architecture*, W. E. Preiser, J. C. Vischer, and E. T. White (eds). New York: Van Nostrand Reinhold.

MacKenzie, I. S. and Isokoski, P. (2008). Fitts' throughput and the speed-accuracy tradeoff. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '08)*. New York: ACM Press, pp. 1633-1636.

Matheson, L. N. (2004). History, design characteristics, and uses of the pictorial activity and task sorts. *Journal of Occupational Rehabilitation 14* (3), pp. 175-195.

Moffatt, K. and McGrenere, J. (2010). Steadied-bubbles: Combining techniques to address pen-based pointing errors for younger and older adults. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '10)*. New York: ACM Press, pp. 1125-1134.

Mustonen, T., Olkkonen, M. and Häkkinen, J. (2004). Examining mobile phone text legibility while walking. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '04)*. New York: ACM Press, pp. 1243-1246.

Newell, A. F. (1995). Extra-ordinary human-computer interaction. In *Extra-Ordinary Human-Computer Interaction: Interfaces for Users with Disabilities*, A. D. N. Edwards (ed). Cambridge, England: Cambridge University Press, pp. 3-18.

Newell, A. F. and Gregor, P. (2000). 'User sensitive inclusive design'—in search of a new paradigm. *Proceedings of the ACM Conference on Universal Usability (CUU '00)*. New York: ACM Press, pp. 39-44.

Persad, U., Langdon, P. and Clarkson, J. (2007). Characterising user capabilities to support inclusive design evaluation *Universal Access in the Information Society 6* (2), pp. 119-135.

Potter, R. L., Weldon, L. J. and Shneiderman, B. (1988). Improving the accuracy of touch screens: An experimental evaluation of three strategies. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '88)*. New York: ACM Press, pp. 27-32.

Pransky, G. S. and Dempsey, P. G. (2004). Practical aspects of functional capacity evaluations. *Journal of Occupational Rehabilitation 14* (3), pp. 217-229.

Price, K. J. and Sears, A. (2008). Performance-based functional assessment: An algorithm for measuring physical capabilities. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '08)*. New York: ACM Press, pp. 217-224.

Ringbauer, B., Peissner, M. and Gemou, M. (2007). From "design for all" towards "design for one"–A modular user interface approach *Proceedings of the 4th Int'l Conference on Universal Access in Human-Computer Interaction (UAHCI '07)*. Berlin: Springer-Verlag, pp. 517–526.

Schwesig, C., Poupyrev, I. and Mori, E. (2004). Gummi: A bendable computer. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '04)*. New York: ACM Press, pp. 263-270.

Sears, A. and Shneiderman, B. (1991). High precision touchscreens: Design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies 34* (4), pp. 593-613.

Sears, A., Lin, M., Jacko, J. and Xiao, Y. (2003). When computers fade... Pervasive computing and situationally-induced impairments and disabilities. *Proceedings of the 10th International Conference on Human-Computer Interaction (HCI Int'l '03)*. Mahwah, New Jersey: Lawrence Erlbaum Associates, pp. 1298-1302.

Sears, A., Young, M. and Feng, J. (2008). Physical disabilities and computing technologies: An analysis of impairments. In *The Human-Computer Interaction Handbook*, J. A. Jacko and A. Sears (eds). New York: Lawrence Erlbaum Associates, pp. 829-852.

Shinohara, K. and Tenenberg, J. (2007). Observing Sara: A case study of a blind person's interactions with technology. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '07)*. New York: ACM Press, pp. 171-178.

Shinohara, K. and Wobbrock, J. O. (2011). In the shadow of misperception: Assistive technology use and social interactions. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '11)*. New York: ACM Press. To appear.

Shneiderman, B. (2000). Universal usability. *Communications of the ACM 43* (5), pp. 84-91.

Smith, R. V. and Leslie, J. H. (1990). *Rehabilitation Engineering*. Boca Raton, Florida: CRC Press.

Smith, S. S. and Kondraske, G. V. (1987). Computerized system for quantitative measurement of sensorimotor aspects of human performance. *Physical Therapy 67* (12), pp. 1860-1866.

Stary, C. (1997). The role of design and evaluation principles for user interfaces for all. *Proceedings of the 7th International Conference on Human-Computer Interaction (HCI Int'l '97)*. Amsterdam, The Netherlands: Elsevier Science, pp. 477-480.

Steinfeld, E. (1994). The concept of universal design. *Proceedings of the Sixth Ibero-American Conference on Accessibility*. Rio de Janeiro (June 19, 1994).

Stephanidis, C., Akoumianakis, D. and Savidis, A. (1995). Design representations and development support for user interface adaptation. *Proceedings of the 1st ERCIM Workshop on User Interfaces for All (UI4All '95)*. Hellas, Crete, Greece: ICS-FORTH.

Stephanidis, C., Salvendy, G., Smith, M. J. and Koubek, R. J. (1997). Towards the next generation of UIST: Developing for all users. *Proceedings of the 7th International Conference on Human-Computer Interaction (HCI Int'l '97)*. Amsterdam, The Netherlands: Elsevier Science, pp. 473-476.

Stephanidis, C., Salvendy, G., Akoumianakis, D., Bevan, N., Brewer, J., Emiliani, P. L., Galetsas, A., Haataja, S., Iakovidis, I., Jacko, J. A., Jenkins, P., Karshmer, A. I., Korn, P., Marcus, A., Murphy, H. J., Stary, C.,

Vanderheiden, G., Weber, G. and Ziegler, J. (1998). Toward an information society for all: An international R&D agenda. *International Journal of Human-Computer Interaction 10* (2), pp. 107-134.

Stephanidis, C. (2001a). Adaptive techniques for universal access. *User Modeling and User-Adapted Interaction 11* (1-2), pp. 159-179.

Stephanidis, C. (2001b). *User Interfaces for All: Concepts, Methods, and Tools*. Mahwah, New Jersey: Lawrence Erlbaum.

Story, M. F. (1998). Maximizing usability: The principles of universal design. *Assistive Technology 10* (1), pp. 4-12.

Terry, M. and Mynatt, E. D. (2002). Side views: Persistent, on-demand previews for open-ended tasks. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '02)*. New York: ACM Press, pp. 71-80.

Trewin, S. and Pain, H. (1997). Dynamic modelling of keyboard skills: Supporting users with motor disabilities. *Proceedings of the 6th Int'l Conference on User Modeling (UM '97)*. Vienna, New York: Springer Wien New York, pp. 135-146.

Trewin, S. and Pain, H. (1999). Keyboard and mouse errors due to motor disabilities. *International Journal of Human-Computer Studies 50* (2), pp. 109-144.

Trewin, S. (2002). An invisible keyguard. *Proceedings of the ACM SIGCAPH Conference on Assistive Technologies (ASSETS '02)*. New York: ACM Press, pp. 143-149.

Trewin, S., Keates, S. and Moffatt, K. (2006). Developing Steady Clicks: A method of cursor assistance for people with motor impairments. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '06)*. New York: ACM Press, pp. 26-33.

Vanderheiden, G. (1998). Universal design and assistive technology in communication and information technologies: Alternatives or complements? *Assistive Technology 10* (1), pp. 29-36.

Vanderheiden, G. (2000). Fundamental principles and priority setting for universal usability. *Proceedings of the ACM Conference on Universal Usability (CUU '00)*. New York: ACM Press, pp. 32-37.

Vogel, D. and Baudisch, P. (2007). Shift: A technique for operating pen-based interfaces using touch. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '07)*. New York: ACM Press, pp. 657-666.

Walker, N. and Smelcer, J. B. (1990). A comparison of selection time from walking and bar menus. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '90)*. New York: ACM Press, pp. 221-225.

Walker, N., Philbin, D. A. and Fisk, A. D. (1997). Age-related differences in movement control: Adjusting submovement structure to optimize performance. *Journal of Gerontology: Psychological Sciences 52B* (1), pp. 40-52.

Wang, J. and Mankoff, J. (2003). Theoretical and architectural support for input device adaptation. *Proceedings of the ACM Conference on Universal Usability (CUU '03)*. New York: ACM Press, pp. 85-92.

Wobbrock, J. O. (2003). The benefits of physical edges in gesture-making: Empirical support for an edge-based unistroke alphabet. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '03)*. New York: ACM Press, pp. 942-943.

Wobbrock, J. O., Myers, B. A. and Hudson, S. E. (2003a). Exploring edge-based input techniques for handheld text entry. *Proceedings of the 3rd Int'l Workshop on Smart Appliances and Wearable Computing (IWSAWC '03). In 23rd Int'l Conference on Distributed Computing Systems Workshops (ICDCSW '03)*. Los Alamitos, California: IEEE Computer Society, pp. 280-282.

Wobbrock, J. O., Myers, B. A. and Kembel, J. A. (2003b). EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '03)*. New York: ACM Press, pp. 61-70.

Wobbrock, J. O., Aung, H. H., Rothrock, B. and Myers, B. A. (2005). Maximizing the guessability of symbolic input. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI '05)*. New York: ACM Press, pp. 1869-1872.

Wobbrock, J. O. (2006). The future of mobile device research in HCI. *CHI 2006 Workshop Proceedings: What is the Next Generation of Human-Computer Interaction?* Montréal, Québec (April 22-27, 2006), pp. 131-134.

Wobbrock, J. O. and Myers, B. A. (2006a). Trackball text entry for people with motor impairments. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '06)*. New York: ACM Press, pp. 479-488.

Wobbrock, J. O. and Myers, B. A. (2006b). Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM Transactions on Computer-Human Interaction 13* (4), pp. 458-489.

Wobbrock, J. O. and Myers, B. A. (2006c). From letters to words: Efficient stroke-based word completion for trackball text entry. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '06)*. New York: ACM Press, pp. 2-9.

Wobbrock, J. O. and Gajos, K. Z. (2007). A comparison of area pointing and goal crossing for people with and without motor impairments. *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '07)*. New York: ACM Press, pp. 3-10.

Wobbrock, J. O. and Myers, B. A. (2007). Enabling devices, empowering people: The design and evaluation of Trackball EdgeWrite. *Disability and Rehabilitation: Assistive Technology 2* (4), pp. 1-22.

Wobbrock, J. O., Fogarty, J., Liu, S.-Y., Kimuro, S. and Harada, S. (2009). The Angle Mouse: Target-agnostic dynamic gain adjustment based on angular deviation. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '09)*. New York: ACM Press, pp. 1401-1410.

World Health Organization. (2009). International classification of functioning, disability and health (ICF). Geneva, Switzerland: WHO Press.

Yamabe, T. and Takahashi, K. (2007). Experiments in mobile user interface adaptation for walking users. *Int'l Conference on Intelligent Pervasive Computing (IPC '07)*. Los Alamitos, California: IEEE Computer Society, pp. 280-284.

Yesilada, Y., Harper, S., Chen, T. and Trewin, S. (2010). Small-device users situationally impaired by input. *Computers in Human Behavior 26* (3), pp. 427-435.