Personalization

Fiona Cisternas

Ability-Based Design: Concept, Principles and Examples

Wobbrock, Kane, Gajos, Harada, and Froehlich

Ability-Based Design

“What can a person do?”

Emphasizing *ability* over *disability*

Ability is *context-dependent*.

Shifting the Burden

- Who is at fault for accessibility issues? User or designer?
- “Adaptation can move the burden of conforming from the human user to the system”

ENABLED

There's already a blueprint for a more accessible internet. If only designers would learn it

By Anne Quito • November 15, 2018

"Many of us are temporarily able bodied and will face exclusion as we age. When we design for inclusion, we're designing for our future selves."

Empathy? (“I’ll never understand…”)

Prior Approaches - Assistive Technology & Rehabilitation Engineering

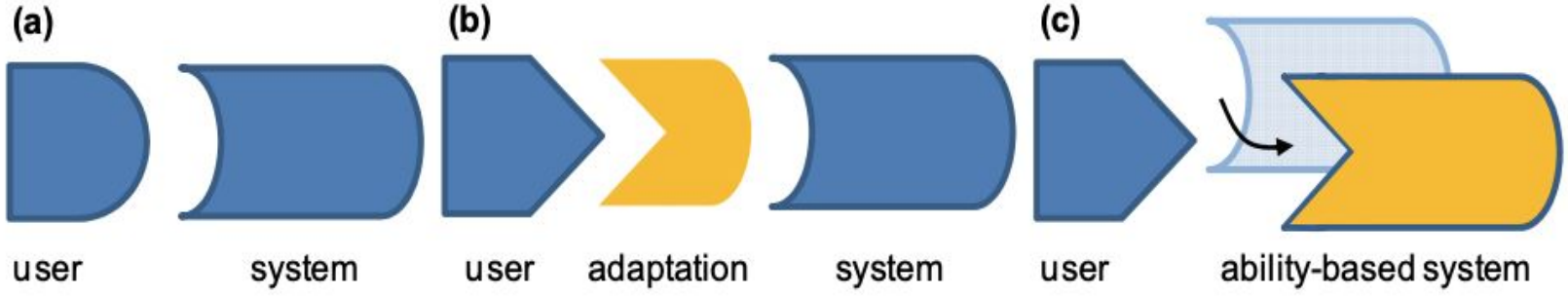


Figure 2. (a) A user whose abilities match those presumed by the system. **(b)** A user whose abilities do not match those presumed by the system. Because the system is inflexible, the user must be adapted to it. **(c)** An ability-based system is designed to accommodate the user's abilities. It may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

Universal Design

What works for almost everyone?

Universal Usability - Universal Design for Interfaces

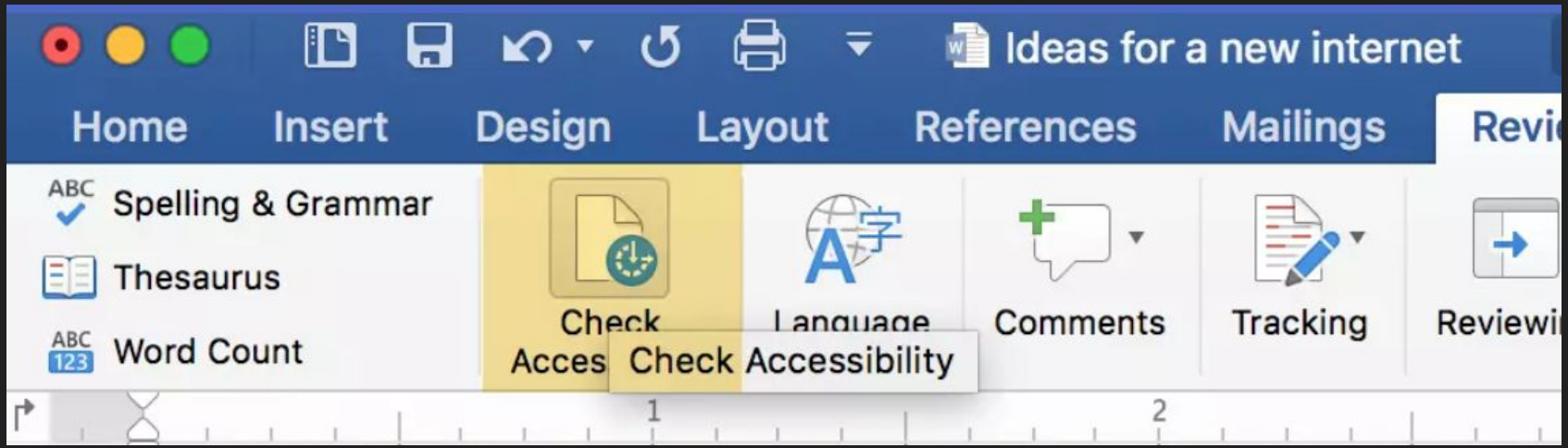
Discussion:

Give an example of a piece of technology that was designed with “universal usability” in mind. What are its benefits and its drawbacks regarding accessibility?

Design for All vs. Design for One

- “What can everyone do?”
- Universal

- “What can you do?”
- Personalized



Principles of Ability-Based Design

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*

Principles of Ability-Based Design

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*

“Of the 14 examples provided (in Table 2) only one cites example of contextual awareness: walking interfaces. This example’s interpretation of context is restricted to “is the user walking or not?”... Context should not merely encompass the current physical activity of the user, but also their mental state. Why do our devices detect when we are walking or running, but not when we are anxious or nervous, when both can be detected with equally (un)reliable models based on the same sensors?”

-Dylan

Principles of Ability-Based Design

Seven Principles of Ability-Based Design

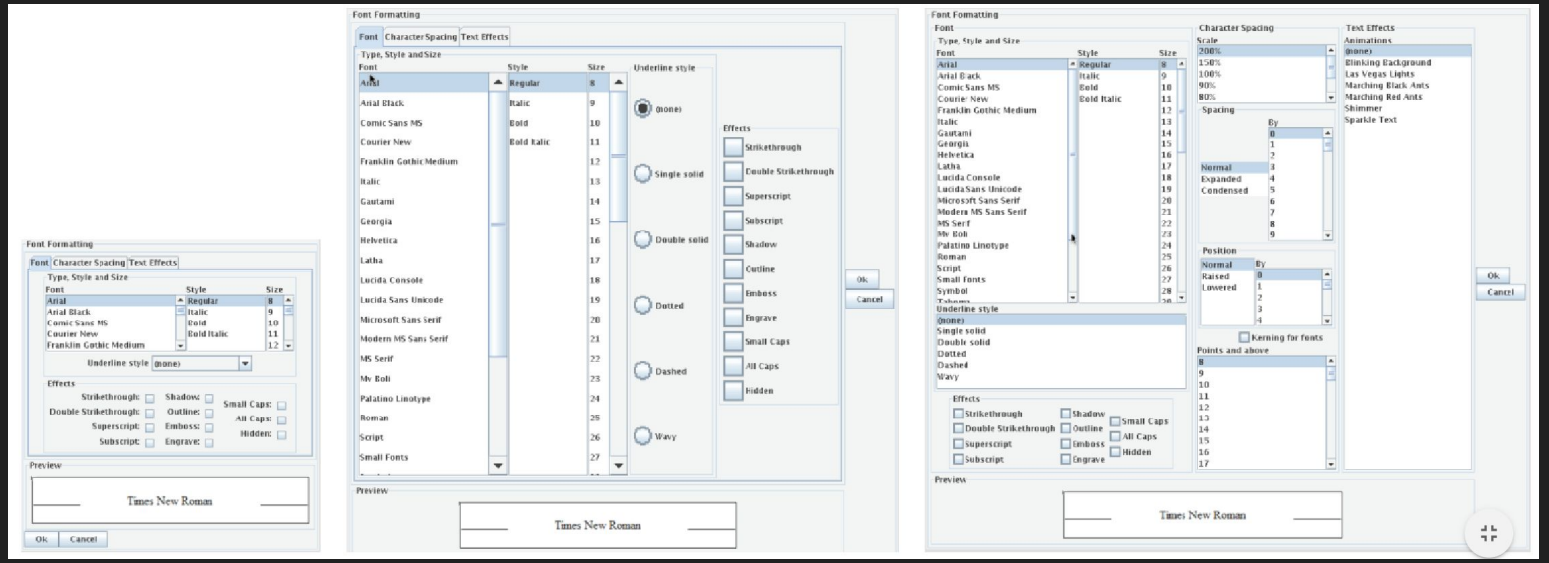
STANCE	**1. Ability.**	Designers will focus on ability not *dis*-ability, striving to leverage all that users *can* do.	*Required*
	2. Accountability.	Designers will respond to poor performance by changing systems, not users, leaving users as they are.	*Required*
INTERFACE	**3. Adaptation.**	Interfaces 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*

Design Principles: Discussion

What are the tradeoffs associated with *transparency* from a design standpoint? Give an example of how this might manifest in an actual UI.

Supple

- User completes an ability assessment battery
- UI automatically adjusts to the user's needs
- Motor-impaired users were 28% faster, 73% more accurate, and significantly more satisfied



Supple: Discussion

If we are able to generate predictive models based on a user's performance, is it better to make local or global adaptations to the interface? What are the trade-offs?

(Question credit to last year's discussion)

“Given that many designs tend to respond to user input, how much can a system offload the burden of change from a user? That is, is it possible to create a system that does not require a training/onboarding session where the data is modeling to create a new interface? Ultimately, while I do think that the burden of change should be placed on a system for truly accessible design, this paper does not entirely help layout how that could happen.”

-Hazel

Conclusion

Ability-based design is the “universal application of design-for-one”, in which all systems are completely tailored to their user’s needs.

Are there drawbacks to this?

Is it attainable?

Beyond Performance: Feature Awareness in Personalized Interfaces

Findlater and McGrenere

Types of GUI Personalization

Adaptive: System-controlled (Supple)

Adaptable: User-controlled (iPhone apps)

Mixed-initiative: Combination of adaptive and adaptable

Discussion: Personalized GUIs

Do intelligent systems “dumb down” the user because cognitive load is offloaded to the system? If yes, is this negative? Why?

Awareness

Does personalizing a user interface negatively impact the user's awareness of its features?

Does personalization lead to more efficient *core task performance* at the cost of less efficient *new task performance*?

Core Task Performance vs. New Task Performance

Design Factors of Personalized GUIs

- Control: Adaptive, Adaptable, or Mixed-Initiative
- Granularity: Coarse-grained or Fine-grained changes
- Visibility of change: Hide, move, resize, replicate, mark
- Frequency of change

Study 1: Marked and Minimal vs. Full Interface

- Different types of interfaces (marked, minimal, full) were used in PowerPoint
- Users had to complete a simpler task using the interface they were assigned (between-subjects) and then a more complex task using the full interface
- Core task completion was better for minimal users, but awareness (via recognition test) of unused features was worse
- Marked condition made very little difference

Six Differences, anyone?

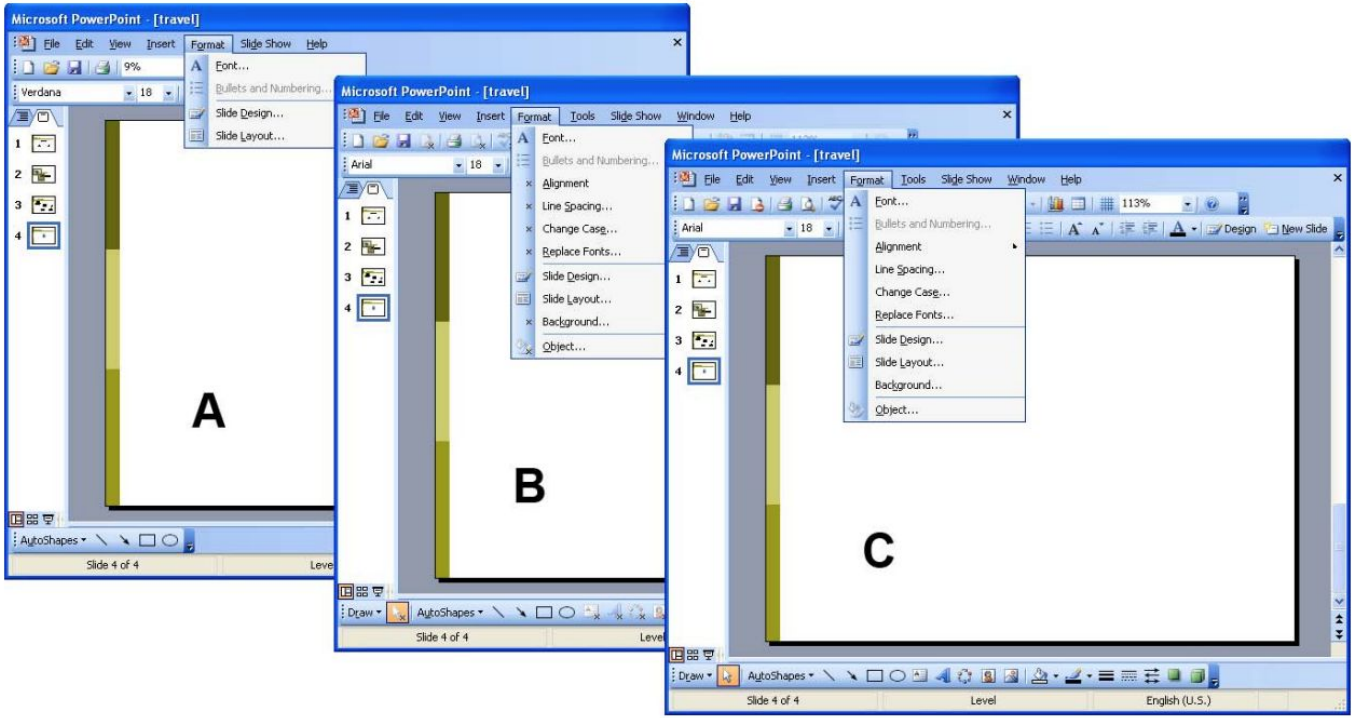


Figure 1. Sample screenshots from the interface layers used in Study 1: minimal interface layer (A), marked interface layer (B), and full interface layer (C).

Study 2: Adaptive Split Menus

- Adaptation controlled by system - rearranging menus to have the most relevant results on top
- No “hiding” involved
- Groups: control, low accuracy (50%), high accuracy (78%)
- Higher accuracy tended to have lower awareness
- Smaller screen size also lowered awareness

Study 3: Impact on New Task Performance

- High/low/control conditions like Study 2, large screens only
- Within-subjects (for statistical power)
- 30 subjects, compensated $10/hr
- Participants familiarized themselves with the interface in the “training block”, then moved on to the “testing block”
- Awareness operationalized as the time taken to select items in the testing block that were not selected in the training block (new task performance)
- Recognition test also administered
- **Confound control:** 3 items were added to menus for the control condition so control menus would not be shorter than other conditions’ menus

Results

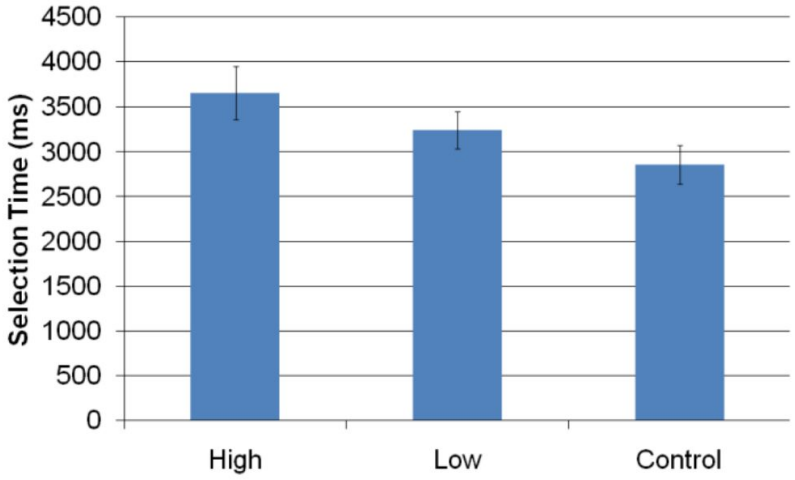


Figure 4. Performance impact of awareness, measured as speed of selecting new items in testing block; 95% confidence intervals shown. (*N* = 29)

"Different levels of awareness have the potential to impact future performance."

Results

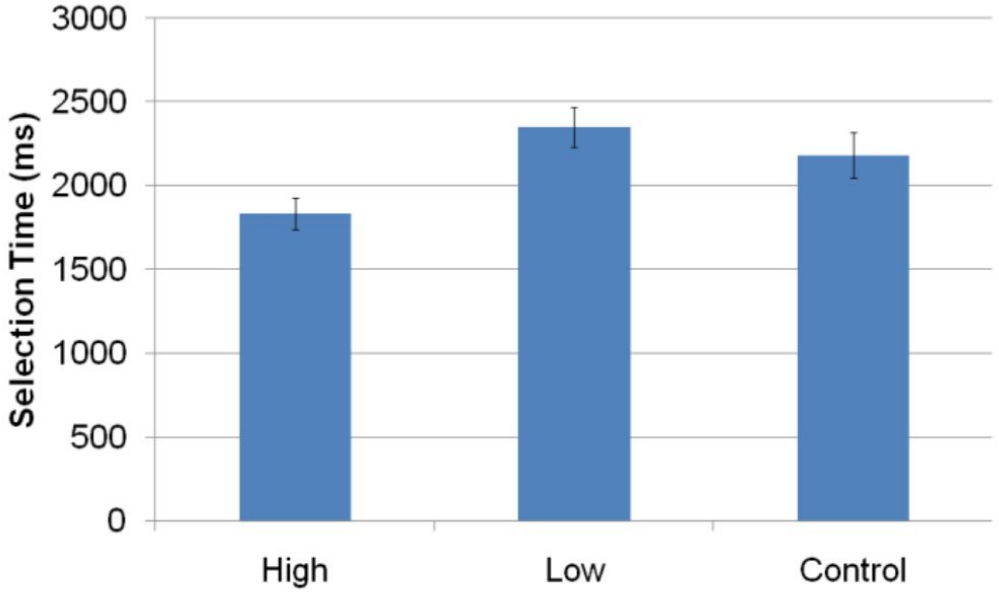


Figure 5. Experienced core-task performance; 95% confidence intervals shown. ($N = 29$)

Conclusions

- Accuracy should not be the ultimate goal of personalization
- Tradeoffs between core performance tasks and awareness should be considered for all design characteristics implemented into the GUI
- Add support for exploring new features
- Enhance discoverability for features not commonly used.
- New features can be manually introduced to the user.

Discussion: Wrap-up

We've been presented with a tradeoff: personalization versus feature awareness. How can we mitigate the issues that arise from this?

Take an everyday UI, like your phone's. Would you rather it prioritize core task performance, or new task performance? Would you rather it be adaptive or adaptable? Why?

Thank you!