

The Ability-Based Design Mobile Toolkit: Developer Support for Runtime Interface Adaptation Based on Users' Abilities

Junhan Kong, Mingyuan Zhong, James Fogarty, Jacob O. Wobbrock

Current Developer Toolkits are *Not* Ability-Aware

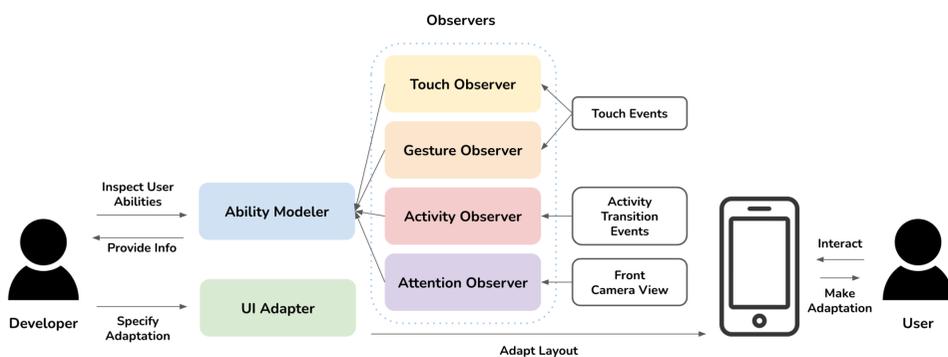
Most current mobile apps are oblivious to their users' abilities, displaying output and reacting to input in ways that generally treat every user the same. For example, a user with limited fine motor control like tremor might have to adapt themselves to tap on small targets accurately, e.g., using a capacitive-tipped pointing stick. Developers who wish to create accessible ability-aware mobile applications face significant challenges, as doing so requires extensive effort to process input and sensor data.

We seek to provide mobile developers a toolkit that supports making apps aware of and responsive to a user's situated abilities at runtime. To this end, we present the **Ability-Based Design Mobile Toolkit (ABD-MT)**. ABD-MT integrates with an app's input and sensor processing to enable developers to observe a user's behavior at runtime, model and reason about the user's abilities, and adapt interface widgets and layouts to better accommodate these abilities.



The Ability-Based Design Mobile Toolkit

The Ability-Based Design Mobile Toolkit (ABD-MT) combines three main concepts to collectively support efficient development of ability-aware applications: the **Observers**, the **Ability Modeler**, and the **UI Adapter**.



Observers

- 1. Capture user interaction data.** The Observers each maintains a history of a user's touches, gestures, activities, and attention.
- 2. Expose human performance metrics.** The Observers allow developers to access a rich set of low-level human performance metrics characterizing user behaviors, e.g., touch direction, variability, movement variability, total duration of each activity or attention.

Ability Modeler

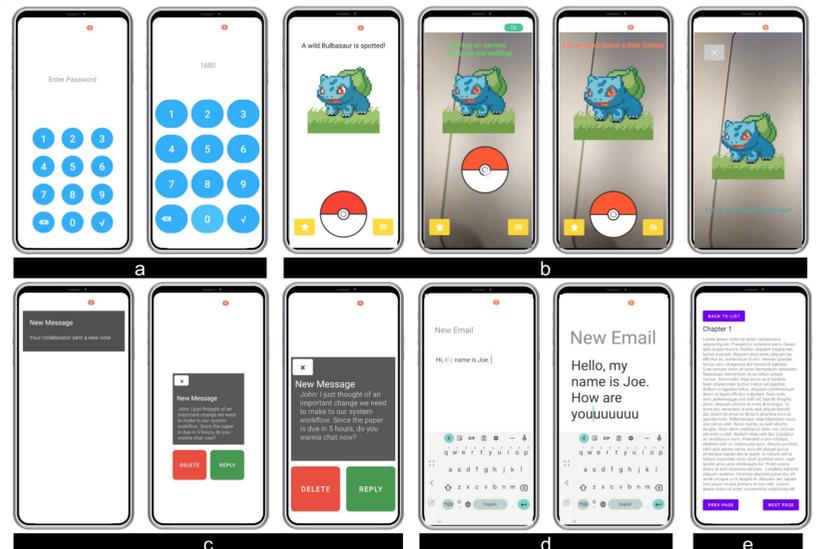
The Ability Modeler takes the observed user behavior as well as the metrics calculated in the Observers and synthesizes these inputs to model user abilities. It provides:

- 1. User ability information for each Ability Group** — fine motor abilities, physical activities, and attention, and
- 2. The interaction of different Ability Groups** — for example, whether observed tremor might be due to walking, or whether a user is looking at their phone while they are doing some physical activity.

UI Adapter

The UI Adapter keeps a record of all user interface (UI) widgets and exposes a set of methods for a developer to manipulate multiple UI widgets, such as changing widget sizes, showing, hiding, and activating widgets, and switching UI layouts.

Example Applications with ABD-MT



- (a) A password keyboard enlarges proportionally to the amount of tremor detected; added buttons and enlarged widgets and fonts;
- (b) An adaptive Pokémon Go game brings targets closer and turns on camera for situational awareness during walking;
- (c) Running-aware notifications responds to walking through
- (d) A zooming text editor makes it easier for users to finely position a text cursor between two characters;
- (e) A reading app changes screen brightness based on attention.

Evaluation Results

We evaluate the toolkit usage with four common methods of toolkit usage evaluation:

Lines-of-Code Reduction

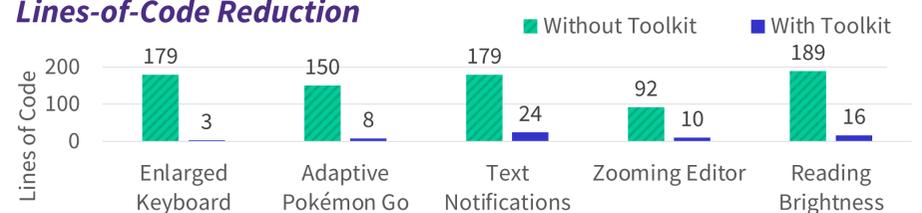


Chart shows the lines of code required to achieve ability-aware functionality.

Usability Study

11 participants with varying levels of Android development experience took part in a lab study, and were asked to make a to-do list app ability-aware under 4 scenarios using ABD-MT. All participants completed the adaptations with an average total time of 23.8 minutes. Participants described the toolkit as “very straightforward,” “very intuitive,” “easy to understand,” and “easy to use.”

Take-Home Study

4 participants spent extra time after the lab study and implemented an ability-aware app of their choice. These include: (1) an adaptive music player for walking, (2) a messaging app that enlarged buttons when walking, (3) a screen time monitoring app based on time looking at display, (4) a to-do list app that hides contents when driving.

Additional Apps Ideated by Participants

Participants proposed 62 unique app ideas that can be built using ABD-MT. Categories include: music and video players (10), maps (8), reading (5), social media (5), messaging (4), games (3), calendar (2), etc.

Acknowledgment. This work was supported in part by Google, by the University of Washington Center for Research and Education on Accessible Technology and Experiences (CREATE), and by National Science Foundation award #IIS-1702751.