

[Yahoo! User Interface Blog \(YUIBlog\)](#)

- [Blog](#)
- [About](#)
- [YUI Jobs](#)
- [Contribute](#)

Site Search:

[Challenges of Interface Design for Mobile Devices](#)

October 2, 2007 at 1:36 pm by Lucas Pettinati | In [Design](#), [Development](#) | [22 Comments](#)

The single most important concept to master when designing mobile device interfaces is “context”. The context in which an application is used and the context of how information is input are both key issues; each must be understood before a well crafted design may be implemented. When these two notions of context are explored, it becomes clear that designing for a mobile device can lead to a solution that is worlds different than its desktop equivalent.

Context of Use

Mobile devices are excellent at connecting users to information; and while consumption of information is likely the largest segment of mobile device usage, interacting with a mobile device to perform important tasks is a usage segment that deserves significant attention. This is because generative work conducted on mobile devices tends to be tactical in nature and demands a sense of immediacy. Users have a very specific need and desire to accomplish their goal in the easiest and fastest way possible. This fact alone helps explain why mobile interfaces are designed the way they are:

- Feature sets are optimized to streamline common use cases
- Use typography to show hierarchy and importance
- Features are progressively displayed
- Large buttons are used to make interactions actionable

Let’s explore a mobile email client as an example of how these attributes are manifested.

Streamline Common Use Cases

Whether you’re lounging at the beach, riding the subway, or at a client meeting, the need to access email on your mobile device is likely predicated by a sense immediacy. If at this point you thought “What about boredom?”, I’d like to remind you of the beach scenario. A complex interaction involving zooming, tiny check boxes, and the like is the last thing one needs. Several mobile email applications address this challenge by displaying an optimized interface that allows users to select an inbox or folder, view a list of messages, and then act on a specific message. Though this model may not be best when dealing with bulk actions, it simplifies the interaction to a primary use case and allows users to get the information they need. Once the user has access to the information being sought, additional options are contextually presented.

Show hierarchy and importance

Instead of using a data table layout with dedicated columns for each piece of email meta data, the iPhone email interface groups sender, date, subject, and message status information into repeating units that make identifying a message a simple task. By varying font size, weight, color, and style, one is able to discretely communicate each morsel of information without having to label it. This not only reduces interface clutter but allows the data to speak for itself.

Features are progressively displayed

While performing bulk actions like flagging several messages is an extremely useful feature, it is probably not a primary activity performed on a mobile device. In the case of the iPhone, options to delete a message, reply, and move to a different folder are only displayed when viewing a message. Taking this concept one step further, tapping the reply icon allows the user to specify if they wish to reply, reply all, or forward the message. By simplifying the mechanism for how one selects a message to a single click, and removing the ability select multiple messages, all features that deal with acting on a message are progressively disclosed in context of that message.

Large buttons are used to make interactions actionable

When you use a laptop or desktop computer, chances are that you're in a very controlled environment; lighting is good, you sit a comfortable distance from the monitor, and using a mouse or trackpad to control a screen cursor is a simple task. In contrast, mobile devices may be used in unpredictable situations; outdoors in very bright light, in the course of another activity (like driving), or while in constant motion—which makes coordinated movements difficult to perform. By making the clickable area of an action large, many of these issues are resolved. When highlighted by a contrasting background color, important actions are more easily seen and targeted even when overall screen contrast is poor. Most important of all, a large click area requires less precision and effort to activate; an excellent manifestation of [Fitts's law](#).

Context of the medium

Interfaces designed for a desktop internet browser experience are usually not optimized for a mobile internet browser. Aside from the issues arising out of the context of use, the mobile device itself may present you with challenges and opportunities not present in the desktop realm:

- Dealing with phone numbers and other mobile friendly data
- Displaying information on a smaller screen
- Not using a cursor
- Device speed and network latency

Dealing with phone numbers and other mobile-friendly data

If your mobile web application displays phone numbers, making it easy for a user to dial that phone number should be a top priority. By using the tel: URI scheme and linking a phone number, many mobile internet browsers will dial the linked phone number.

Some mobile devices may redirect certain URLs to native applications. This is the case with the iPhone redirecting YouTube and Google Maps URLs to the onboard application, but can also be observed with phones that automatically open the built-in calendar when an iCal file is downloaded, or open the built-

in address book when a vCard file is downloaded. To design an experience that can gracefully coexist with others tools, one needs to understand what kind of media can be processed by specific mobile internet browsers, and when onboard applications are launched.

Displaying information on a smaller screen

While most desktop web applications are designed for 800×600 or 1024×768 resolutions, mobile devices employ significantly smaller displays. Some mobile browsers scale graphics to fit their screen width and some mobile browsers allow the user to scale and magnify a portion of the interface. Using traditional web development techniques of creating fluid designs that scale horizontally is the fastest way to deploy a single design to many different mobile devices.

Older mobile devices may have a horizontal resolution of as few as 96 pixels while newer models have horizontal resolutions of 240 or 320 pixels. If you are bothered by the wide range of horizontal resolutions and feel that your design options are discretely different for different resolutions, think about creating resolution-specific CSS files. Common horizontal resolutions are 96, 128, 176, 240, and 320 pixels.

If you have the luxury of targeting specific devices, don't overlook the option of serving device-specific CSS files. Doing this allows you to fine-tune object dimensions for a specific device as well as customizing your application's skin to provide an onboard application feel, if so desired.

Not using a cursor

Mice, joysticks, scroll-wheels, keypads, fingers, and styli—oh my! Different mobile devices require different tools to interact with their interfaces. While each of these input tools accomplishes the essential task of selecting an object, each also presents us with limitations.

Joysticks, scroll-wheels, and keypads typically use up/down commands to scroll content and automatically focus on fields, links, and buttons. In essence, this limits a designer and developer to listening for focus, blur, and click events. Advanced devices that use fingers and/or styli as the input tool may provide access to more events, but here too a wide range of device specific behavior may restrict event listening to a small set. Blazer and PocketIE browsers, depending on device configurations, may allow a user to scroll a page by using a stylus to interact with scroll bars; the same browser on a different device may be limited to a joystick input tool. The Safari browser on the iPhone offers page scrolling by allowing the user to push the page in the direction of the scroll. Whereas Blazer and PocketIE browsers could allow for the opportunity for direct object manipulation (like drag and drop), such an action is not natively available on the iPhone because the act of dragging is reserved for a system-level interactions.

All is not lost, however. Just because a device does not support dragging or double clicking does not mean it has a poor event model. The act of dragging an object to a drop area may seem trivial—but when this action needs to be performed in a moving subway while clinching a hand strap, the mechanics involved can range from being frustrating (using one's thumb to drag and other fingers to hold device) to outright masochistic (trying to control a stylus from moving around a slippery surface).

Device speed and network latency

Until the day that WiFi and WiMAX technologies are built into every mobile device, planning on data transfers via GPRS and EDGE networks is a reality—which means data is transferred as slow as 60

kbits/sec. Even if one is able to overcome the speed and latency of a slow data network, there's still the mobile device's processor and RAM to contend with. As expected, a desktop or laptop computer is several times more powerful than a typical mobile device. Complex code execution on a mobile device may therefore be several times slower than the same code run on a desktop computer. More importantly, mobile devices are all about the consumption of information; lightweight interfaces (both in terms of features and bytes) deliver information faster.

What it boils down to

Simplifying interactions and streamlining access to information may not be as cool as crafting richer interactions, but they are the most effective techniques available at producing a design that is easy to use on a mobile platform. Simplified interfaces can also lead to leaner code, which means less data is transferred over slow connections.

Designing with awareness to context will yield a more atomic design that instead of introducing users to a proverbial blank canvas, will guide them toward accomplishing important tasks. Having to deal with slow data speeds, high network latency, smaller screens, and an unpredictable mode of use only reinforce the need to isolate an application's essential features and offer access to them when contextually appropriate.

Next time you design an interface for a mobile device, remember to consider context of use and context of the medium as part of your design strategy.

Online Resources

The links below provide more in-depth information about several popular mobile platforms:

ACCESS / Palm OS

- [ACCESS | Palm OS Application Design](#)
- [Zen of Palm \(PDF\)](#)

Apple iPhone

- [Apple iPhone Human Interface Guidelines](#)

BlackBerry

- [Optimizing Content for the BlackBerry Browser \(PDF\)](#)

Nokia

- [Nokia Web Browser Design Guide](#)

Microsoft Mobile Internet Explorer

- [Microsoft | Mobile Web Development](#)

Motorola