Malmo University Advanced Interaction Design program application

Sergiu Sebastian Tauciuc

Design decisions

I. Scope and posture

Scope – unite the palmtop and the mobile phone

Besides the laptop, Ronny currently uses palmtop and mobile phone in his work. His goals could be better served with a single device that integrates palmtop and mobile phone capabilities. In a busy life such as Ronny has, it is less efficient to use two devices that have inter-related functions and often serve the same goals, when these goals could be achieved with a single device that could unite these related functions. In this instance, replacing two devices with just one is also necessary for a simple reason: handling. Ronny is a busy man, and handling two devices in the same time (the need could easily occur) is just cumbersome. For Ronny the (possibly) added complexity will be acceptable, as long as it makes a big impact in efficiency.

Posture – sovereign and transient

The device will have to take two different postures in its interaction, depending on context:

Transient posture

A mobile phone has a transient posture. We take it out of our pocket, easily make a call, talk, end the call and put it back in the pocket. In its role as a communication tool, the device will have to adopt a transient posture. Calling and messaging will be always readily available and easy to achieve.

Sovereign posture

Since it will offer features like web-browsing and document viewing/editing, where Ronny's attention is captured for longer periods of time, the device (or the applications running on it) must also be able to adopt a sovereign posture, where all of the screen reel-estate is used and the interaction is more discrete.

II. Form, handling, input methods

Form: non-folding, non-sliding, large display, keyboard on screen

To maximize screen reel-estate, the **qwerty 'keyboard'** will be displayed on-screen, and only used when needed.

The device will have a '**landscape' default layout**, which better supports two-handed handling. For viewing documents or 'portrait' photos, the display will be able to change its layout based on how the device is held (using accelerometers)

Size: to be discussed.

Size should be discussed with a industrial designer. The paper prototypes are considering the following approximate sizes:

width: 11cm

height: 8cm

thickness: 1,5cm

Handling

hold with 2 hands

use stylus and head-set

make accessible without stylus (through keys + touch)

> Having to talk on the phone a big part of the day, Ronny will use a wireless headphones and microphone set. This enables him to have both his hands available to use the device (or perform other actions) while talking. By holding the device in his left hand and using a pressure pen with the right hand, Ronny can perform advanced operations while in the middle of a conversation.

Input methods and physical controls

microphone, speaker (for use without head-set)

head-set

stylus

on-screen keyboard

touch-sensitive sides on the top and right for scrolling

On/Off button

Volume Down, Volume Up buttons

Mute All / Un-mute button

Ok / Call button

Cancel / End Call button

Navigation buttons (arrows: Left, Right, Top, Down)

‘home’ button

Small led to signal missed calls or new messages.

> The touch-sensitive sides will enable scrolling long lists with the fingers when the stylus isn’t used. This is useful especially when manipulating the contacts list, which must allow as fast as possible access. Scrolling by sliding the finger on the sides could also prove useful during normal navigation (web, images), because using the stylus is a bit cumbersome for scrolling. To be tested with users for confirmation and

optimization!

The *Volume Down*, *Volume Up* and *Mute All* buttons will start on the front of the device and continue on the bottom side (see prototype). Thus, the conversation volume buttons will be accessible when the device is used without the head-set, and the ***Mute All*** button will be accessible and easily-recognizable inside a pocket (for turning the sound off easily - instead of rejecting - when a call is disturbing a meeting and call forwarding isn't enabled).

The rest of the buttons are positioned on the front of the device, near the bottom side (see prototype).

The microphone will be positioned towards the left side of the device and the speaker towards the right side – so when Ronny won't be using the headset, he will be holding the phone with the right side up and the left side down.

III. Interaction idioms

Windowed applications, scrolling

The device will use windows to contain different actions or running programs, from calls to web browsing. Windows will have 2 modes:

- **small size mode** will use the first available space found on the screen, on the center stage (see prototypes)
- **maximized mode** will use the entire screen space, pushing the 'fixed' components (Applications and Contact List, see below) to the sides, and allowing their recall on demand.

Scrolling will be used both for the main screen and for applications running maximized. Since the scroll-by-touch mechanism allows intensive and easy scrolling, we can afford to organize content vertically, within certain limits. The trade-off might come in aesthetics, so scroll-bar graphics should be made discrete enough as to not affect the aesthetics of the main screen.

IV. Default View

The default view has 3 fixed components (see paper prototype):

- Applications
- Missed Calls and Messages / History
- Contact List (Address Book)

All these components can be 'minimized' to occupy only a fraction of their default size, bound to their corresponding edge, and then either used from their minimized form or maximized again.

The **Contact List** sits on the left side of the screen and resembles an IM friend list.

For each contact, it offers different ways of communication, depending on the contact data available. The default action (done with *double-tap* or *select and Enter*) is regular phone-call.

Unlike an IM friend list, the contact list has a Search component at the top. This is normal for our context, considering the usual size of phone address books. The search component is filter-as-you-type, narrowing the contacts with each key type.

Also unlike an IM list, the contact list has 2 display modes:

The default mode shows the entire list of contacts in alphabetical order, and finding a certain contact is done through either scrolling or filtering (searching).

The optimized mode, the one Ronny will use, will show only the most frequently or/and recently used 6 contacts in the address book[1], which will fit into the first screen (no scrolling needed), so Ronny can access them immediately.

In this mode, “pinning” will also be available. For example, Ronny can pin his family members at the top of the list, so that their position in the list is consistent, indifferent of how often he speaks with them in certain periods.

Navigation in the contact list can be done using the keys, in a similar way as current models: pressing the down key in the default view selects the first contact in the list and so on. Thus the *up* and *down* keys can be used for navigating between contacts and the *Call* button can be used for calling.

The **Applications** are all on the right side, in order of estimated importance. When an application is open, it takes the entire space available on the ‘desktop’, which is by default empty (only has a background image).

When an application is closed, the state of the application is saved, so that the next time it is open it will start from where it was left.

The **Missed Calls and Messages** list will be empty by default, containing only the link to (Call) History.

As new messages or missed calls are received, they are added to the list, with basic details available like *caller/sender*, *time* and the *first lines of the message* if it is a message.

The items in the list are expandable in-line, with a larger preview presented. A *Read in Inbox* link is available for messages – this could be needed in the case of larger SMS-s or e-mails.

The items also have a *Close* button, which removes them from this list. Thus, irrelevant items can be removed from the list without necessarily having to be opened first.

Pressing the **Home** physical button will close any open applications (except ongoing conversations) and return to the default view.

V. Prototypes

A few paper prototypes have been added for better communicating parts of the solution and reducing the length (and understandability) of this document.

The **device prototype** (in which the buttons are explained) is a stand-alone.

[1] Research has shown that, in average, 90% of a user’s phone-calls are directed to 5-6 people in their phone-book.

The **other prototypes** should be used as support for the key-path scenarios, which have as purpose the validation of the solution against the context scenarios.

VI. Notes

1. My first solution to this assignment was given 2 years ago (unfortunately, although admitted, I was unable to start the program at that time). Since then, new technologies and products have become available (or at least I have learned about them), so I thought an adaptation would be in order. Also, I thought the initial solution needed a few revisions. I hope the new solution is better.

3. The device prototypes presented are re-used paper. That does not mean any disregard for the high-standard and requirements of the evaluation process, but rather two personal convictions:

- prototypes are throwaways by their nature. The "back side" doesn't matter as long as they serve their communication purpose
- paper, as other resources, should be saved and reused as much as possible.

2. This solution is by no means final or complete and does not intend to be. It could be, at best, a starting point for a design-build-test-redesign process.

VII. References

[1] Cooper, A., Reinman, R., *About Face 2.0: The Essentials of Interaction Design*, 2003

[2] Goodwin, K., *Getting from Research to Personas: Harnessing the Power of Data* http://www.uiconf.com/7west/goodwin_article_2.htm, 2002

[3] Saffer, D., *Designing for Interaction: Creating Smart Applications and Clever Devices*, 2006