Interaction Design
for
Small Screen Layouts

Lauri Sumari
<Lauri.Sumari@iki.fi>

5 May 2004

University of Art and Design Helsinki
Department of Industrial and Strategic Design

User Interface Design, Spring 2004

Abstract

Screen size and quality, variability of contexts of use and usage patterns are often very different in smart products and traditional PC GUIs. This document lists some general interaction design guidelines specifically for small screens. The daunting task of designing a usable small screen interface can be achieved through economical use of screen space, new interaction methods, and guiding the user with landmarks and animational clues. Varying lighting conditions, the tight coupling between the screen and the keyboard, and the pronounced impact of internationalisation must also be taken into account. Personalisation, extensibility and adaptivity can further help overcome the challenges posed by the small screen.

Introduction

"Smart products" or "information appliances" are very different in many respects from traditional PCs. One of the important differences is a smaller and lower quality display screen.

The price, durability and size requirements all make it more difficult to provide a high quality display in a smart product. Pixel resolutions of these screens are typically an order of magnitude smaller than in the PC realm. Instead of 1280 x 1024 pixels, we may have 128 x 128 pixels, often even less. Similarly, the individual pixels are often larger, resulting in a blocky image. Finally, the number of colours supported by the screen is usually smaller. Even monochrome screens are common in smart products, whereas a contemporary PC screen is able to display millions of colours.

There are differences between smart products, of course. Obviously a digital camera, for example, needs a much better display than the average mobile phone. The general size constraint still holds in most cases, however.

Also, small screens are often encountered in mobile products, where the context of use may vary in time, even for the same user. Fieldwork and truly mobile use, where the product is actually used while walking, running, or crawling, may require interaction models that are very different from those used in stationary or merely portable products.

Furthermore, different interaction modes may be needed for different contexts of use of a single product. For example, the maintenance person hanging up a telegraph pole looking up technical data from a smart product needs to use a different interaction mode than when sitting back in the van recording the mileage for the servicing trip. The observer crawling among bushes taking notes of animal behaviour has similar requirements. The businessperson looking up a meeting from the calendar during a call needs to access the product's functions in a way that is fundamentally different from the way the product is used when no call is in progress. The screen layout must reflect these differences.

Interaction Design Guidelines

The small screen places new requirements for interaction design. One cannot simply take a PC GUI an put it into a smart product. The small screen will only be able to present a small amount of information at a time. Therefore, when designing a GUI for that kind of screen, one must take great pains to make sure that the information presented is as relevant as possible, and that it is presented in the most useful way.

As the discipline of small screen UI design is relatively young, there is an ongoing effort to define good guidelines. Established usability guidelines, such as those presented by Nielsen (1993), probably still hold in theory, but are too broad to take into account the specific requirements for smart products. There are also specific guidelines for small screen UIs. For example, Kärkkäinen and Laarni (2002) present several guidelines for designing web sites for small screen use.

In this article, however, the approach is that of designing the screens of *information appliances*, as discussed by Norman in (Bergman, 2000). In the information appliance paradigm, the aim is to design a specialised, often small and mobile, product to satisfy certain user needs related to a set of tasks that naturally belong together. In an information appliance, one can't design screen layout properly without taking into account not only the task and context of use, but also the input hardware, such as buttons, and the form factor. The UI of the whole product must be designed as a whole. Therefore, guidelines for UIs that may be viewed on a variety of small screens are not sufficient here. This article attempts to compile some general guidelines for designing the screen layouts and interaction techniques of an information appliance.

Here is a list of the guidelines:

1 Less is definitely more
2 Fundamentally non-PC interaction techniques are needed
3 Remember varying viewing conditions
4 Don't forget the keys
5 Consider internationalisation early on
6 Use varying layouts to help communicate location within the interface
7 Use animational clues to help communicate the interaction model

The current wording of some of these guidelines may strike one as slogan-like or provocative. Hopefully, that at least serves to make them more memorable. This is a work in progress.

1 Less is definitely more

The popular usability slogan *less is more* is true for larger screens also, to a great extent. For small screens, minimalisation and simplification are absolutely crucial, because screen real estate is at a premium.

Also, the usage of smart products may be more goal-oriented than the usage of a PC sometimes is. After all, a smart product is design for an activity of limited scope — that is the whole point of information appliances. A PC, on the other hand, is very versatile, and a large number of unrelated applications is common. The user does many things in front of the screen, and sometimes may "surf the web", for example, at a leisurely pace, not looking for anything in particular. He may even find himself looking for something to do!

When using a smart product, the user typically has a distinct goal, so the interaction must be focused and efficient (Kärkkäinen and Laarni, 2002). Cluttered layouts must be rigourously avoided to keep the interaction simple.

Find the most important actions. Task analysis should be used to form a clear understanding of the frequencies of use of each function.

Challenge every element's right to be visible on the top levels. To eliminate clutter and have only the important things in the important places, every UI element's inclusion must clearly justified. This is especially important on the highest levels of the UI.

Don't decorate in space. Decorating UI elements often requires more space to be done properly, if only because of the large proportion of screen space a single pixel represents. Even simple decorations like border lines may require too much space, because a border usually needs white space on both sides to work (Bergman, 2000). "Decorations in time", such as animational clues are eligible, however (see guideline "Use animational clues to help communicate the interaction model" below).

Every pixel counts. Because a single line or column of pixels may represent several percent of screen space, every pixel should be carefully allocated. For example, increasing the font height of a row of text by a single pixel may, on a screen full of text, have a drastic effect on the amount of information that can be presented at a time.

2 Fundamentally non-PC interaction techniques are needed

The usage pattern of a smart product is often very different from the one found in PC use. Transient usage is typical. There are often short bursts of activity (for example, checking the location for a meeting), between which the product may remain unused for long periods of time. On a PC, one often performs tasks that take longer to complete, because more complex tasks are possible. For example, on a smart product with a small screen one would not write a 10-page report, or draw complex diagrams and charts. That's what the PC is for.

Also, as the tasks themselves are different, the kinds of interaction techniques optimal for them are likely to be different as well. Finally, the different physical constraints may also affect the choice of interaction techniques, size being the most obvious example. Certainly, the differences in usage pattern, tasks and physical constraints are interrelated. They all suggest, however, that one should not limit oneself to only considering interaction techniques prevalent in the PC domain, but other, even completely new techniques should also be considered. For example, scroll bars work quite well on large screens, but small screens may benefit more from a different approach (see Björk and Redström (1999) for an example of how to give the user a better overview of a large document by replacing scroll bars with flip zooming).

Don't just copy the PC GUI. Solutions that supposedly work well in PC GUIs are likely to be less than optimal in a small screen GUI. Don't copy solutions simply for the sake of consistency and familiarity.

Be creative with possible solutions. No single solution exists to all problems, and at any rate the solutions and widgets that are currently most common, may not be optimal. Sometimes, one may even have to invent new widgets and interaction techniques to find the best solution. Nevertheless, much research has been done in the field of information visualisation, and methods already exist that could help improve the interaction and presentation models of small screen GUIs.

Caveat: Familiarity and consistency should be leveraged when possible. There is wisdom the principle of adhering to established standards. Leveraging the familiarity of PC GUI elements is desirable, and should be done when feasible. Sometimes, however, one needs to depart from the familiar and find a solution more suitable for the smaller screen.

Some examples of less widespread interaction techniques that could facilitate small screen use:

– *Flip Zooming* (see for example Holmquist, 1997 or Björk and Redström, 1999) is an example of a *focus + context visualisation technique,* in which an overview (context) and a detailed view (focus) are presented at the same time using zooming and possibly distortion (Figure 1). Other such techniques include the *generalised fisheye view* (Furnas, 1986), *Continuous Zoom* (Bartram et al., 1995), *Perspective Wall* (Mackinlay et al., 1991), and *Document Lens* (Robertson and Mackinlay, 1993).

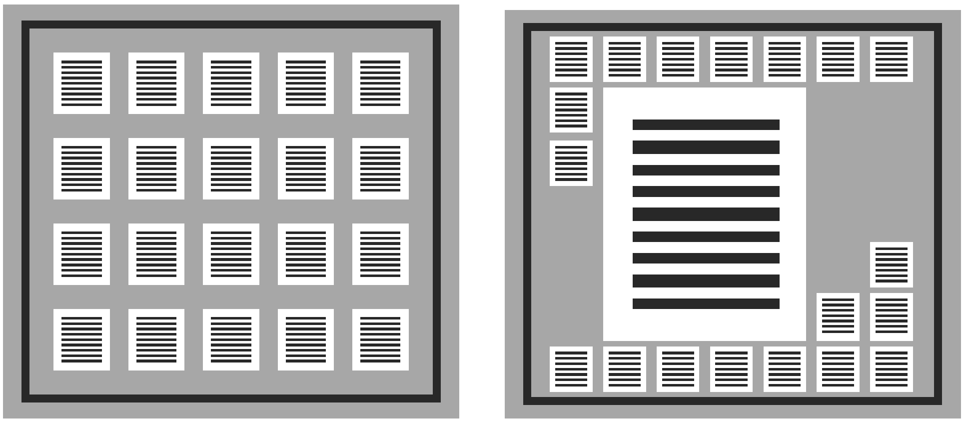

Figure 1. Principle of Flip Zooming (from Holmquist, 1997)

– *Exposé* <URL: http://www.apple.com/macosx/features/expose/> is an interaction technique implemented by Apple Computer for quickly moving between a number of open windows and the desktop in its PC operating system, Mac OS X. A key press or mouse gesture is used to activate the function which then arranges all open windows and shows them in a reduced size allowing easy and fast navigation to any of them (Figure 2). This technique resembles the focus + context visualisation techniques.

Figure 2. Exposé (from Apple Computer, Inc., 2003)

– *Dynamic presentation methods* present information sequentially in time instead of space. Automatically scrolling text is an example of such a method.
Transparent widgets could be used to overlay buttons and menus on top of the content so that the content would still be visible from underneath.

– *Automatic hyphenation* of text could decrease the need to scroll by allowing more efficient use of individual lines of text. It could also make reading easier by reducing the large variation between line lengths that short lines typically introduce.

– *Automatic or dynamic summarisation* of text could help condense long texts on the fly to better fit the small screen. This method was used in combination with Flip Zooming by Björk and Redström (1999) to offer a "keyword view" that allows the user to see the big picture even though not all text fits on the screen.

3 Remember varying viewing conditions

Small screens often aren't continuously lit. To save battery power, most smart products only light up the screen when the product is in use, if at all. Still, some of the screens must be readable only at a glance, even when the screen is viewed off-axis, from a distance, or in bad light. Sometimes a screen needs to light up to be more clearly readable and noticeable from afar — for example, the user needs to be alerted to an incoming call when a phone is in silent mode.

Task analysis must determine which screens are likely to be viewed differently. To ensure effortless viewing of information whenever possible, varying viewing considerations must be taken into account.

4 Don't forget the keys

A small screen often has accompanying buttons nearby, the design of which affects the screen layout design to some extent, and vice versa. So-called *softkeys* — keys whose label is displayed on the screed next to the key — are especially tightly coupled with the screen. Where softkeys are used, screen layout and key layout become interrelated.

Key layout and screen layout needs to be addressed together. Because designing the screen layout and key layout depend upon each other, they can no longer be thought of as entirely separate entities, as they can in PCs.

Caveat: Users may still view the screen and keys as distinct entities. It has been shown in user studies that softkeys are sometimes problematic. Users may not be accustomed to viewing the screen and the keys as having a direct relationship (Lindholm and Keinonen, 2003).

5 Consider internationalisation early on

Word lengths vary greatly across different languages. As their length may drastically affect the choice of layout and navigation models, it is critical to address the question of which languages to support early in the product design process. Internationalisation can no longer be applied on top of a finished product, as it can be (if the product is well engineered) in large screen GUIs.

Supported languages required must be chosen carefully. The set of supported languages must be balanced to allow for a clean screen design.

Term choices in each language must take into account the small screen. In addition to being appropriate to users, terms used in the UI must be chosen for their brevity in order to save space.

6 Use varying layouts to help communicate location within the interface

Small screen size easily results in deep menu structures or nested dialogues. The user should be provided with visual landmarks to help keep track of one's current location within the GUI. If all menus and dialogues look identical, users will get lost and have a hard time finding anything or remembering the way back to previously visited places.

Vary the layout of different modes to help differentiate between them. For example, different levels of the menu hierarchy might have visually different layouts. Too monotonic layouts look boring and are easy to get lost in.

Caveat: Don't abandon consistency. Again, consistency is still important to bear in mind in order to avoid a chaos of different looking screens.

7 Use animational clues to help communicate the interaction model

The use of animational clues is useful even in desktop size layouts, although it is not used very much (The GUI of the Mac OS X operating system by Apple Computer is an exception, and indeed puts animational clues into excellent use). As decoration is impossible in space due to the size constraint, designers are hard pressed to find ways to visualise UI elements so as to provide a good affordance. Furthermore, some of the interaction models used may be new and initially hard to grasp without assistance or experience.

Animate actions in a way that reflects their nature in the interaction model. By using a transitional animation tailored to give visual hints about the nature of the transition between different modes, instead of just snapping from one layout to another, the designer can make it easier for the user to understand the interaction model. For example, when a folder is opened, its contents may be made to visually emerge from within the folder. This can help novice users unfamiliar with the concept of a folder to understand the GUI more easily.

Caveat: Don't overdo it. Don't allow animation to distract the user's attention or slow down the interaction. Keep the animations brief, only use them where they provide a clear benefit, and avoid persistent animations.

Optimisation possibilities

Finally, there are some considerations which may not apply to all smart products, but which may, in certain situations, help tackle the screen size problem. Personalisation, extensible design and adaptive solutions can sometimes help optimise the trade-off between size and features.

Personalisation

Allowing the user to do the final optimisation of the GUI can help cater to the needs of a wide audience. That optimisation is relevant because any possibility to eliminate unneeded GUI elements or to reorganise the elements in a way that better servers the user is valuable in a situation where screen space is tightly constrained. "Less is more", when applied by the GUI designers, always involves trade-offs. This will leave someone aching for the ability to add a shortcut to that one feature that would be so much nicer to have on the top level, or to the ability to replace a less important element with it. The ability to personalise may help reach the level of simplification and effectiveness that is impossible to provide with a single, uniform solution deployed across a wide range of users.

Consider possibilities to support personalisation of the GUI. Try to give a chance for the user to fine-tune the GUI to show the most relevant things on top and in the most appropriate fashion possible.

Caveat: Personalisation GUI may add complexity just by virtue of existing. Personalisation needs some facility to control which personalised changes to apply. It may do more harm than good by adding complexity to accommodate the personalisation facility itself. On the other hand, this facility may be placed outside the GUI of the small screen product — it may make sense to do the personalisation on a PC, for example.

Extensibility

To keep the product simple, one needs to keep the core feature set minimal. Sometimes, however, adding a single function, service, or application would be very useful — to some users, at least. The product might be just what the user needs, except it lacks, say, a calculator. This suggests a modular, extensible design. The Handspring Visor PDAs, for example, have built-in slots, into which extension modules may be inserted to add new functionality to the unit (Bergman, 2000). If such an extensible architecture is used, the screen layout must to be able to cope with changing functionality.

Small screen GUI's should consider allowing addition of new elements. A new element may be a button to activate the new service or application, for example.

Caveat: An extension GUI may add complexity just by virtue of existing. Again, if the extension facility itself requires a special GUI to handle, it may not be worthwhile at all. It must be completely automated, or at the very least moved out of the smart product itself.

Adaptivity

The small screen may often force the user to do serial search, because information can be presented only a little bit at a time. Thus, it is a good idea to look for ways to shorten or eliminate such tedious, time consuming searching. Adaptive, "intelligent" GUI solutions can help improve defaults to make it more likely that the desired choice is among the first options visible (see Sumari (2003) for a discussion of making menus faster to use with an adaptive approach). This can shorten the search if done properly.

Adaptivity can also be used to condense detailed information to better suit the poor screen size and resolution (see Sarjakoski and Nivala (2003) for an example of how to present maps in a small space using adaptive methods).

Consider using adaptive UI solutions to condense and suggest. Like personalisation, adaptivity may help overcome the challenge of UI element prioritisation dynamically. Adaptive solutions should be considered when designing for small screens.

Caveat: Adaptivity may be too slow or consume too much power with current technology. Adaptivity often involves machine learning, which may be computationally too expensive for use in current mobile products. This may change in the future.

Caveat: Adaptivity is hard! An attempt to use adaptivity to improve usability may actually deteriorate it, because adaptivity tends to undermine consistency (among other things; see Pylkkänen (2003) and Annala (2003) for discussions of adaptive solutions' usability challenges). Dynamical changes in the interface may be too confusing. This may also change in the future, once the field matures.

References

Annala, R. (2003). *Älykkäiden käyttöliittymien käytettävyysvaatimukset.* Salovaara, A. et al. (eds.). Perspectives on intelligent user interfaces, pp. 19–35. Espoo: Otamedia. ISBN 951-22-6458-7 Available online at: <URL: http://www.soberit.hut.fi/publications/ReportSeries/Reports/HUT-SoberIT-C1.pdf>

Bartram, L. et al. (1995). *The Continuous Zoom: A constrained fisheye technique for viewing and navigating large information spaces.* Proceedings of the ACM, UIST 95, pp. 207–215. New York: ACM Press.

Bergman, E. (ed.). (2000). *Information appliances and beyond — Interaction design for consumer products.* London: Academic Press. ISBN 1-55860-600-9

Björk, S. and Redström, J. (1999). *An alternative to scrollbars on small screens.* CHI '99 extended abstracts on Human factors in computing systems, pp. 316–317. New York: ACM Press ISBN 1 -58113-158-5

Furnas, G. W. (1986). *Generalized fisheye views.* Proceedings of the ACM, SIGCHI '86, pp. 16–23. New York: ACM Press.

Holmquist, L. E. (1997). *Focus+Context visualization with Flip Zooming and the Zoom Browser*. CHI 97 Electronic Publications.
Available online at: <URL: http://www.acm.org/sigchi/chi97/proceedings/poster/leh.htm>

Kärkkäinen, L. and Laarni, J. (2002). *Designing for small display screens.* Proceedings of the second Nordic conference on Human-computer interaction, pp. 227–230. New York: ACM Press. ISBN 1-1-58113-616-1
Available online at: <URL: http://www.soberit.hut.fi/publications/ReportSeries/Reports/HUT-SoberIT-C1.pdf>

Lindholm, C. and Keinonen, T. (eds.) (2003). *Mobile usability — How Nokia changed the face of the mobile phone.* New York: McGraw–Hill Professional. ISBN 0-07-138514-2

Mackinlay, J. D. et al. (1991). *The Perspective Wall: Detail and context smoothly integrated.* Proceedings of the ACM, SIGCHI '91, pp. 173–179. New York: ACM Press.

Nielsen, J. (1993). *Usability engineering.* London: Academic Press. ISBN 0-12-518406-9

Pylkkänen, H. (2003). *Älykkään käyttöliittymän lähestymisen ongelmat.* Salovaara, A. et al. (eds.). Perspectives on intelligent user interfaces, pp. 5–18. Espoo: Otamedia. ISBN 951-22-6458-7
Available online at: <URL: http://www.soberit.hut.fi/publications/ReportSeries/Reports/HUT-SoberIT-C1.pdf>

Robertson, G. G. and Mackinlay, J. D. (1993). *The Document Lens.* Proceedings of the ACM, UIST '93, pp. 101–108. New York: ACM Press.

Sarjakoski, L. T. and Nivala, A. (2003). *Context-aware maps in mobile devices.* Salovaara, A. et al. (eds.). Perspectives on intelligent user interfaces, pp. 112–133. Espoo: Otamedia. ISBN 951-22-6458-7
Available online at: <URL: http://www.soberit.hut.fi/publications/ReportSeries/Reports/HUT-SoberIT-C1.pdf>

Sumari, L. (2003). *To adapt or not to adapt — A case study: the user interface of a heart rate monitor.* Salovaara, A. et al. (eds.). Perspectives on intelligent user interfaces, pp. 144–149. Espoo: Otamedia. ISBN 951-22-6458-7
Available online at: <URL: http://www.soberit.hut.fi/publications/ReportSeries/Reports/HUT-SoberIT-C1.pdf>