CHAPTER IV

4. Interaction Design Basics

4. Interaction Design Basics

Interaction design is about creating interventions in often complex situations using technology of many kinds including PC software, the web and physical devices.

- Design involves:
  - achieving goals within constraints and trade-off between these
  - understanding the raw materials: computer and human
  - accepting limitations of humans and of design.
- The design process has several stages and is iterative and never complete.
- Interaction starts with getting to know the users and their context:
  – finding out who they are and what they are like . . .probably not like you!
  – talking to them, watching them.
- Scenarios are rich design stories, which can be used and reused throughout design:
  – they help us see what users will want to do
  – they give a step-by-step walkthrough of users' interactions: including what they see, do and are thinking.
- Users need to find their way around a system. This involves:
  – helping users know where they are, where they have been and what they can do next
  – creating overall structures that are easy to understand and fit the users' needs
  – designing comprehensible screens and control panels.
- Complexity of design means we don't get it right first time:
  – so we need iteration and prototypes to try out and evaluate
  – but iteration can get trapped in *local maxima*, designs that have no simple improvements, but are not good
  – theory and models can help give good start points.

4.1 Introduction

Some of HCI is focused on understanding: the academic study of the way people interact with technology. However, a large part of HCI is about doing things and making things – design.

So, interaction design is not just about the artifact that is produced, whether a physical device or a computer program, but about understanding and choosing how that is going to affect the way people work. Furthermore, the artifacts we give to people are not just these devices and programs, but also manuals, tutorials, online help systems.

4.2 What is design?

A simple definition is: achieving goals within constraints

This does not capture everything about design, but helps to focus us on certain things:

- **Goals**
- **Constraints**
- **Trade-off**

4.2.1 The golden rule of design

This leads us to the *golden rule of design*: understand your materials

For Human–Computer Interaction the obvious materials are the human and the computer. That is we must:

- understand *computers*
  – limitations, capacities, tools, platforms
- understand *people*
  – psychological, social aspects, human error.

4.2.2 To err is human

In fact, the opposite is the case: physical materials are treated better in most designs than people.

The phrase 'human error' is taken to mean 'operator error', but more often than not the disaster is inherent in the design or installation of the human interface. Bad interfaces are slow or error-prone to use. Bad interfaces cost money and cost lives. People make mistakes.

Often when an aspect of an interface is obscure and unclear, the response is to add another line in the manual. People are remarkably adaptable and, unlike concrete lintels, can get 'stronger', but better training and documentation are not a panacea. Under stress, arcane or inconsistent interfaces will lead to errors.

4.3 The process of design

If you design using a physical material, you need to understand how and where failures would occur and strengthen the construction, build in safety features or redundancy.

4.3.1 The central message – the user

We will discuss the information on basic psychology, on particular technologies, on methods and models.

This is the core of interaction design: put the user first, keep the user in the center and remember the user at the end.

- **Requirements – what is wanted**

- **Analysis -** These techniques can be used both to represent the situation as it is and also the desired situation.
- **Design -**Well, this is all about design, but there is a central stage when you movefrom what you want, to how to do it.
- **Iteration and prototyping** Humans are complex and we cannot expect to get designs right first time. We therefore need to evaluate a design to see how well it is working and where there can be improvements.
- **Implementation and deployment** Finally, when we are happy with our design, we need to create it and deploy it. This will involve writing code, perhaps making hardware, writing documentation and manuals – everything that goes into a real system that can be given to others.

4.4 User focus

As we've already said, the start of any interaction design exercise must be the intended user or users. This is often stated as: know your users.

So, how do you get to know your users?

- Who are they?
- Probably *not* like you!
- Talk to them.
- Watch them.
- Use your imagination

4.5 Navigation design

The design would make sense to the user and also that proposed implementation architectures would work. In addition scenarios can be used to:

- **Communicate with others** – other designers, clients or users.
- **Validate other models** A detailed scenario can be 'played' against various more formal representations such as task models or dialog and navigation models.
- **Express dynamics** Individual screen shots and pictures give you a sense of what a system would look like, but not how it behaves.

This linearity has both positive and negative points:

- **Time is linear:** Our lives are linear as we live in time and so we find it easier to understand simple linear narratives.
- **But no alternatives:** Real interactions have choices, some made by people, some by systems. A simple scenario does not show these alternative paths. In particular, it is easy to miss the unintended things a person may do.

Scenarios are a resource that can be used and reused throughout the design process: helping us see what is wanted, suggesting how users will deal with the potential design, checking that proposed implementations will work, and generating test cases for final evaluation.

But now we are focusing on the computer system itself. You interact at several levels:

- **Widgets** The appropriate choice of widgets and wording in menus and buttons will help you know how to use them for a particular selection or action.

- **Screens or windows:** You need to find things on the screen, understand the logical grouping of buttons.
- **Navigation within the application:** You need to be able to understand what will happen when a button is pressed, to understand where you are in the interaction.
- **Environment** The word processor has to read documents from disk, perhaps some are on remote networks. You swap between applications, perhaps cut and paste.

The structure of an application is to think about actual use:

- Who is going to use the application?
- How do they think about it?
- What will they do with it?

This can then drive the second task – thinking about structure.
Here we will consider two main kinds of issue:

- Local structure
  – looking from one screen or page out
- Global structure
  – structure of site, movement between screens.

4.5.1 Local structure

Much of interaction involves goal-seeking behavior. Users have some idea of what they are after and a partial model of the system. However, in a world of partial knowledge users meander through the system.

The Levels of interaction

PC application

- Widgets
- Screen design
- Navigation design
- Other apps and operating system

Physical device

- Buttons, dials, lights, displays
- Physical layout
- Main modes of device

The real world!

- Website
- Form elements, tags and links
- Page design

Site structure

- The web, browser, external links

Here are four things to look for when looking at a single web page, screen or state of a device.

- Knowing where you are
- Knowing what you can do
- Knowing where you are going – or what will happen
- Knowing where you’ve been – or what you’ve done.

The screen, web page or device displays should make clear *where you are* in terms of the interaction or state of the system.

It is also important to know *what you can do* – what can be pressed or clicked to go somewhere or do something. Some web pages are particularly bad in that it is unclear which images are pure decoration and which are links to take you somewhere.

You then need to know *where you are going* when you click a button or *what will happen*. Of course you can try clicking the button to see. In the case of a website or information system this may mean you then have to use some sort of ‘back’ mechanism to return, but that is all; however, in an application or device the action of clicking the button may already have caused some effect.

4.5.2 Global structure – hierarchical organization

We will now look at the overall structure of an application. This is the way the various screens, pages or device states link to one another. One way to organize a system is in some form of hierarchy.

This is typically organized along functional boundaries (that is, different kinds of things), but maybe organized by roles, user type, or some more esoteric breakdown such as modules in an educational system.

The hierarchy links screens, pages or states in logical groupings.

4.5.3 Global structure – dialog

In a pure information system or static website it may be sufficient to have a fully hierarchical structure, perhaps with next/previous links between items in the same group. However, for any system that involves doing things, constantly drilling down from one part of the hierarchy to another is very frustrating. Usually there are ways of getting more quickly from place to place.

In HCI the word ‘dialog’ is used to refer to this pattern of interactions between the user and a system.

Human–computer dialog is just the same; there are overall patterns of movement between main states of a device or windows in a PC application, but the details differ each time it is run.

A simple way is to use a network diagram showing the principal states or screens linked together with arrows. This can:

- show what leads to what
- show what happens when
- include branches and loops
- be more task oriented than a hierarchy.

4.5.4 Wider still

Each sits amongst other devices and applications and this in turn has to be reflected within our design.

This has several implications:

- **Style issues** We should normally conform to platform standards, such as positions for menus on a PC application, to ensure consistency between applications.
- **Functional issues** On a PC application we need to be able to interact with files, read standard formats and be able to handle cut and paste.

- **Navigation issues** We may need to support linkages between applications, forexample allowing the embedding of data from one application in another, or, ina mail system, being able to double click an attachment icon and have the rightapplication launched for the attachment.

4.6 Screen Design and Layout

We have talked about the different elements that make up interactive applications, but not about how we put them together. A single screen image often has to present information clearly and also act as the locus for interacting with the system.

The basic principles at the screen level reflect those in other areas of interaction design:
Ask What is the user doing?
Think What information is required? What comparisons may the user need to make? In what order are things likely to be needed?
Design Form follows function: let the required interactions drive the layout.

Tools for layout

We have a number of visual tools available to help us suggest to the user appropriate ways to read and interact with a screen or device.

Decoration

Other decorative features like font style, and text or background colors can be used to emphasize groupings. See how the buttons differ in using the foreground and background colors (green and gold) so that groups are associated with one another. Seealso how the buttons are laid out to separate them into groups of similar function.

Alignment

Alignment of lists is also very important. For users who read text from left to right, lists of text items should normally be aligned to the left. Numbers, however, should normally be aligned to the right (for integers) or at the decimal point. This is because the shape of the column then gives an indication of magnitude – a sort of mini histogram.

White space

In typography the space between the letters is called the counter. In painting this is also important and artists may focus as much on the space between the fore ground elements such as figures and buildings as on the elements themselves.

Entering information

Some of the most complicated and difficult screen layouts are found in forms-based interfaces and dialog boxes. In each case the screen consists not only of information presented to the user, but also of places for the user to enter information or select options. Actually, many of the same layout issues for data presentation also apply to fields for data entry.

Knowing what to do

Some elements of a screen are passive, simply giving you information; others are active, expecting you to fill them in, or do something to them. It is often not even clear which elements are active, let alone what the effect is likely to be when you interact with them!

It is important that the labels and icons on menus are also clear. Again, standards can help for common actions such as save, delete or print. For more system-specifications, one needs to follow broader principles. For example, a button says 'bold': does this represent the current *state* of a system or the *action* that will be performed if the button is pressed?

Affordances

These are especially difficult problems in multimedia applications where one may deliberately adopt a non-standard and avant-garde style. How are users supposed to know where to click? The psychological idea of *affordance* says that things may suggest by their shape and other attributes what you can do to them: a hand leaf fords pulling or lifting; a button affords pushing.

Presenting information

The way of presenting information on screen depends on the kind of information: text, numbers, maps, tables; on the technology available to present it: character display, line drawing, graphics, virtual reality; and, most important of all, on the purpose for which it is being used.

We have an advantage when presenting information in an interactive system in that it is easy to allow the user to choose among several representations, thus making it possible to achieve different goals.

Aesthetics and Utility

Ideally, as with any well-designed item, an interface should be aesthetically pleasing. Indeed, good graphic design and attractive displays can increase users' satisfaction and thus improve productivity. However, beauty and utility may sometimes be at odds.

The conflict between aesthetics and utility can also be seen in many 'well designed' posters and multimedia systems. In particular, the backdrop behind text must have low contrast in order to leave the text readable; this is often not the case and graphic designers may include excessively complex and strong back grounds because they look good.

Localization / internationalization

If you are working in a different country, you might see a document being word processed where the text of the document and the file names are in the local language, but all the menus and instructions are still in English. The process of making software suitable for different languages and cultures is called *localization* or *internationalization*.