

The human ability to learn is characterised by the creation of a link with what is already known, and the transference of this knowledge to new situations. This principle is important as it means that the designer must know the intended target audience for a product when selecting its structure, in order to ascertain what likely prior knowledge and experience the potential user can call upon.



The **context-sensitive cursor** is a good way to give the user specific navigation hints or instructions. However, this feature is only possible if there is a pointer element for navigation.



The **metaphor** is a practical form of the mental model. The metaphor, which is reflective of the human learning, starts with the familiar and transfers this to a new environment. The metaphor forms a narrative framework in which to place the possibilities within the system into a context that is logical for the user. Objects and actions are often borrowed from the real world to illustrate similar functions of the system.

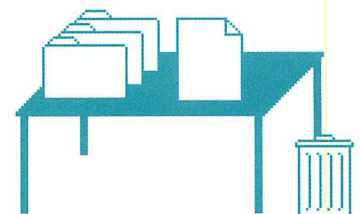
The metaphor of the desktop, for example, enabled computers with this graphical user interface to be used for the first time by people with little or no knowledge of computers. The transfer of everyday objects and processes from the classical office environment and on to the interface allowed users to draw conclusions about the functions and the type of interaction that they offered. Equally, the interface of a pocket PC is closely based on structures that are recognisable from larger PCs, and as such it connects with the mental model that is already familiar to the user.

Metaphors can be a disadvantage. The metaphor as a closed narrative framework has not become established, and nor has a separate, all-embracing metaphor for small-screen devices. Instead, a number of single metaphorical interface elements have become established, such as the button, the folder, the magnifying glass or the dustbin. This is partly due to the fact that closed metaphors are difficult to scale, but a larger problem is the naturalism of metaphors: many functions of a digital system do not have any corresponding function in the real world, as such no metaphorical counterpart can be found.

The **dustbin** is often used as a metaphor for the 'delete' function. This metaphor is so easy to understand that it has been in use for 30 years. It is sensible, especially in a drag-and-drop environment, because it embodies the instruction for interaction: if a file is dragged on to this symbol, it is deleted. It is also possible for the user to change his mind, because files can also be retrieved from the dustbin.



The concept of files that can be saved in folders, which can then be used to create new filing structures, has also become established in many systems. An **enclosing metaphor**, which defines the idea of the working environment in greater detail would actually be a hindrance, because it would transfer real space restrictions into virtual reality.

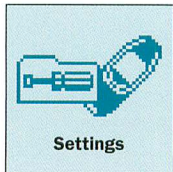


The success of computer-assisted systems would be unthinkable without the use of icons. The 'graphical user interface' (GUI), made the use of computers possible for non-specialists. The desktop metaphor and its associated icons established a recognised standard, which was representative of quite complex information exchanges between the system and the user. Symbolic actions of the user, such as putting something in the dustbin, translates what are actually complex information transactions into understandable concepts, as they are borrowed from real life.

Icons

Using icons on small screens is useful, and justifiably popular. Even if they are not logically connected by a narrative framework, icons still allow fast non-verbal communication between the system and the user. However, designers must develop icons carefully if they are to be commonly understood. For functions that were conceived in the digital arena, such as email, the icons that are used to indicate them will often draw on the established associations of their pre-digital form.

Using the principle of **redundant coding** means that the extra picture caption helps the user to interpret the icon. (Nokia 6310)



For an icon to communicate successfully, the metaphorical image and its meaning in a digital context must be learned first. To ensure this, redundant coding is used: the icon is supported by a textual explanation, which the user can refer to if they are in doubt of the icon's meaning. With use of the device the icon itself will offer enough information for the user to understand. An additional option is to incorporate animated icons; this method can be used to communicate more complicated information and actions.

Icons often incorporate **elaborate design detail**, because of the need to make the interface easily discernable from others; but these details can make the interface harder to understand. Originality in the choice of design and the analogy may backfire if the user does not understand the symbols used.

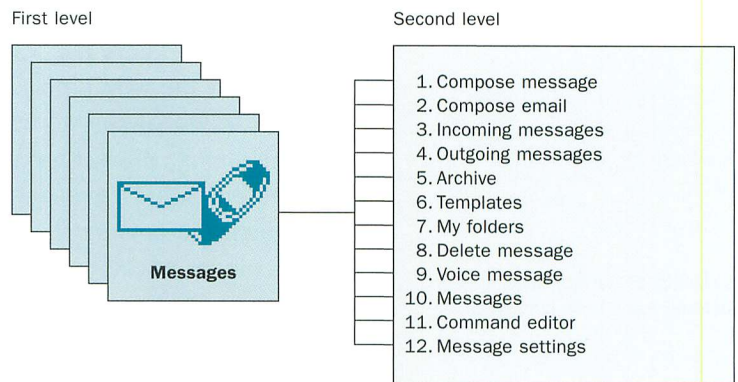


All stages of abstraction can be used in the design of icons. If the space and the graphical capabilities of the device permit, photographic images or sophisticated illustrations can be used. Equally, the smaller the amount of available space and the more limited a device's colour scheme is, the more archetypical the icons should be.

Icon alphabets work across language barriers; they form the traffic signs of a digital application that the user looks for when they wish to navigate a program. To avoid confusing the user and to achieve visual consistency, all icons within a system should have the same degree of abstraction.

The gap between **common sense and originality** defines the window of opportunity for the design of icons. Depending on the recipient and the application, the metaphorical image used may vary. The designer must take into account whether or not the target group consists of users who are experienced with digital media. The type of user will be a crucial factor in the decision of whether an imaginative metaphor can be used, or if the icon should be more conventional and easy to understand.

For most applications, only the top level of the hierarchy and its corresponding tools are described by icons. If the user then chooses to navigate lower levels of the hierarchy, the additional options are displayed as text descriptions. Icons therefore, are limited in their descriptive capability; once a certain degree of complexity is reached, icons become ambiguous and lose their advantages over text descriptions.



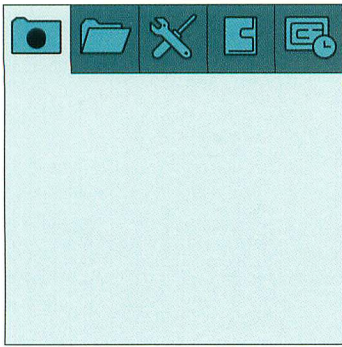
The structure of a device's **navigation levels** will often determine icon use. At the highest level the categories can be represented by icons, but at the secondary level the selection choices are too specific to be unambiguously represented by an icon. (Nokia 6310)

Basic design exercise

Develop an icon that represents the function 'find a restaurant'. This icon should have the capacity to be used at various display sizes and resolutions.

Consider whether an animated icon could help to communicate the content more clearly.

For further examples see Section B in this chapter.



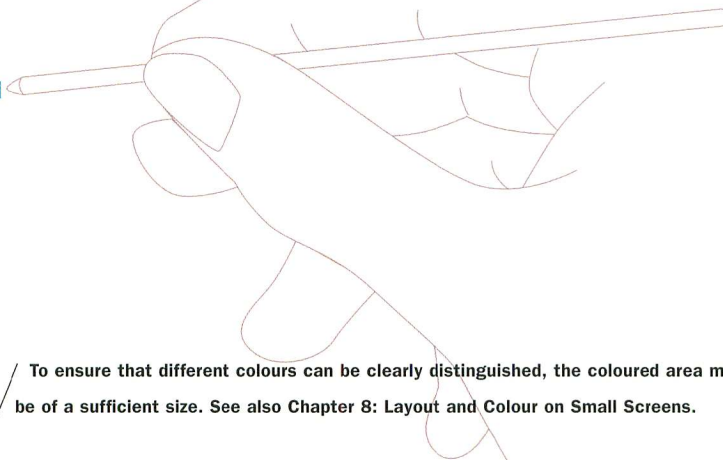
Used as a label on a **row of tabs**, an icon is remembered both by its appearance as well as its location.

In the quest for the optimum use of available space on a small screen, the **size of icons** is a distinct advantage. Icons can offer a great deal of information that simply could not be displayed in text form in the same space. The size of an icon is directly connected to the user's form of interaction with the device, for example, if a stylus is used for input, the element for selection can be displayed at a very small size.

Also, the arrangement of the icons on the screen display supports a user's spatial memory capacity. This means that functions can be found faster because the user remembers both the icon and its position. This applies to the main navigation levels such as those used in many mobile phones and in the navigation and function bars of Pocket PCs.

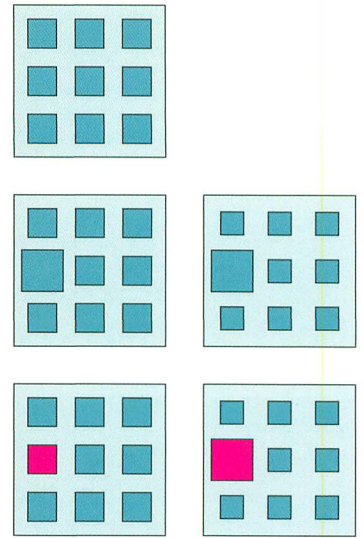
Direct input with a stylus changes the design criteria for icons, as an icon can be selected very accurately. This means that icons can even be displayed at the very small size of 9x9 pixels. For icons of this size to be legible, it is necessary to use: strong contrasts, very simplified forms and no antialiasing. However, colour coding at this size can be used, but only to a limited extent because the space available is just not large enough to distinguish the colours. It would be just about possible to use the two signal colours of red and green because they are distinctive enough even on small spaces.

Stylus-controlled interfaces allow the **smallest icons** to be displayed.



To ensure that different colours can be clearly distinguished, the coloured area must be of a sufficient size. See also Chapter 8: Layout and Colour on Small Screens.

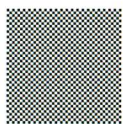
For those devices with **indirect interaction by scroll wheel or jogdial** the icon also serves as a pointer device to tell the user where they are within the system. In order for this to work effectively the icon variations must be visibly changed in its colour, size or contrast to clearly show which option is currently active, and the icon must be allowed sufficient space for this change to be clearly visible. If the change is also to be highlighted by an alteration in size, then the necessary space for the maximum display size must be reserved around the icon. Icons are typically between 16–32 pixels in size.



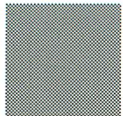
Each icon must have different **display states** developed: active, inactive and selected, so that the user will know where they are in the system

Direct interaction with a touchscreen is the most space consuming technique. Icons that represent the options on a touchscreen must be large enough to be selected by a fingertip, and this is usually a larger area than an icon would need for visualisation. Displays with touchscreens therefore have a much lower density of addressable interaction elements. Increasingly, to circumvent this limitation, touchscreens work with gesture recognition: the user carries out a metaphorical movement, which the system understands as icon input.

Interactive elements on touchscreens should have a **physical size** of at least 15x15 mm, and the distance between the elements should be at least 5 mm.



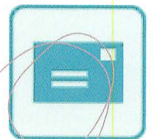
15 x 15 mm at 72dpi > 43 x 43 pixels



15 x 15 mm at 110dpi > 66 x 66 pixels



15 x 15 mm at 144dpi > 86 x 86 pixels

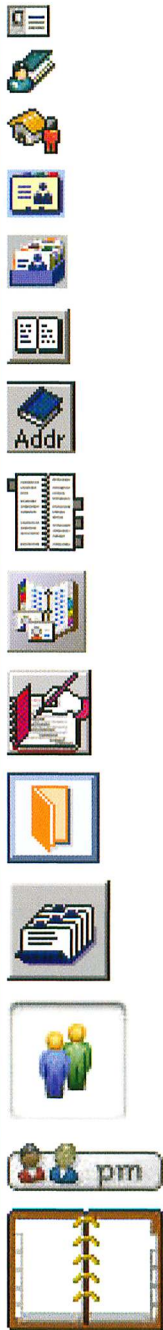


The advent of colour displays that have increasingly high resolutions, means that it is now possible for the visual appearance of screen interfaces to be clearly distinguished. Here, the **design of icons** plays an important role.

Mail



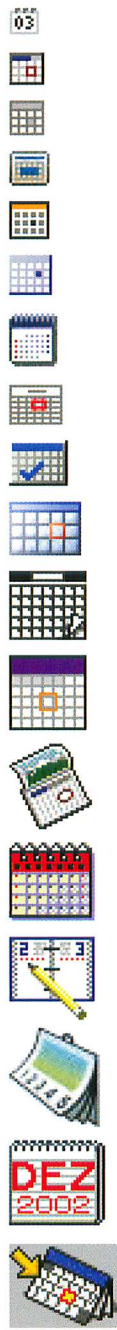
Address book



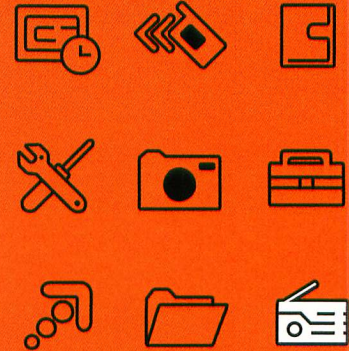
Home



Calendar



Settings



V401SA (Vodafone, Sanyo): carefully developed symbols do not need colour to differentiate them or to achieve a high visual quality. As such, colour can instead be used to indicate activation.



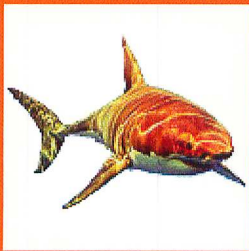
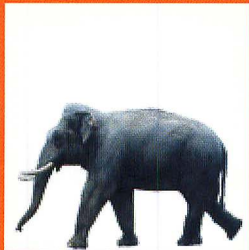
Z200 (Sony Ericsson, 2003): all icons in a software interface should possess the same degree of abstraction and the same visual language. This is not always possible, so the formal link is often created by a three-dimensional presentation style.

Icons should always be developed in at least two states: passive and active. In this example, in addition to a change in the image, the activation is also indicated by a change in size and a frame placed around the icon.

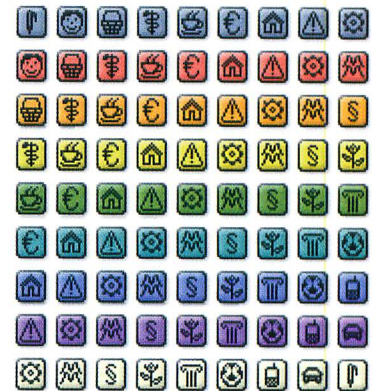
If the available space and resolution quality allows, icons can be designed with a greater degree of detail, which means that they will have an effect on the overall visual appearance. When selecting the icon analogy to use, the experience and background of the target-user group must be taken into account.

Design Exercise in Interface Design:

this is an example of the development of an icon by step-by-step abstraction of a photographic illustration. Apart from the selection of a suitable photograph, further corrections are usually necessary to achieve an archetypical image of the object. The smaller the icon to be developed, the fewer individual features the motif can contain. (Zhang Chao, Franziska Langbrandtner, Julia Ellrich, Alexander Gessler, 2004)



Dining: if animated icons are used, even more complex statements are possible. Here the topic of 'dining' is visualised, the function of which proves to be a more time-consuming or leisurely option than simply 'eating'. (Anna Zesewitz, 2003)



Activis Icon Alphabet: the more extensive the alphabet of icons, the simpler and more striking the individual icons should be in their design. An icon should ideally work in black and white; colour can then be used as an extra coding element, for example, to combine icons into groups. (Hendrik Rieß, 2005)

Dynamic Change in Size: with better resolution quality, the icon can also be dynamically scaled, which helps to make selection easier. This principle can also be transferred to the menu selection on small screens. (Dock function, Apple Mac OSX)



Personal electronic communication is a mixed format that is positioned between formal written communication and casual oral communication. The difficulties of text input have favoured the spread of alternative **image-based language forms**. These enable complex content, such as feelings and emotions, to be communicated in just a few simple input actions.



Icons (au by KDDI, 2005): this example shows the variety of icons offered by the Japanese network operator au by KDDI. To provide the user with even more icon options, tips are given on how to combine several icons in order to visualise even more complex content.



