Topic 4: Design Principles

1. Learnability

Learnability principles are concerned with interactive system features, which aid novice users to learn quickly and also allows steady progression to expertise. The principles discussed below support the learnability design principle.

Simplicity

The simpler the interface, the easier it is to use. Your design should make simple,common tasks simple to do, communicating clearly and simply in the user's own language, and providing good shortcuts that are meaning-fully related to longer procedure.

Predictability

This interactive design principle requires a user's knowledge of interaction to be sufficient to determine the outcome of present or future interaction with the system. One form of the predictability principle is concerned with a user's ability to envisage which operations can be performed next. This is often referred to as operation visibility, and is concerned with showing the availability of operations which can be performed next by the user. Based on this principle, if an operation can be performed then there should be a clear indication of this to the user.

Familiarity

The familiarity principle is concerned with the ability of an interactive system to allow a user to map prior experiences, either real world or gained from interaction with other systems, onto the features of a new system.

A recycle bin is a familiar item in the real world and recycle bin icon immediately suggests its function.

This type of familiarity is also referred to as the guessability of features in the system. Another example of this is the similarity between the computer keyboard and that of a typewriter.

The appearance of an object provides familiarity with its behaviour. Effective use of appearance in an interactive system design can improve the familiarity of the system.

Generalisability

This interactive design principle provides support for users to extend knowledge of specific interaction within, and across applications, to new, but similar situations. For example, cut/copy/paste operations within Microsoft Office applications use of same short-cut keys.

Similarly if a user knows how to draw a rectangle using a drawing package then the user can apply this knowledge to draw a circle using either the same package or other similar packages.

One of the purposes of standards, and programming style guides, is to increase generalisability across a variety of applications.

2. *Flexibility*

Flexibility in interactive design extends the way a user and the system exchange information. By applying flexibility principles to an interactive system design, designers aim to improve a system's usability.

Dialog initiative

When the system controls the dialog flow, the dialog is said to be system preemptive. Conversely, when the flow is controlled by the user, the dialog is said to be user preemptive. In general a user preemptive dialog is favoured although some situations require a system preemptive dialog. In reality some line between these two extremes is usually the most satisfactory solution.

Multi-threading

Within a user interface a thread can be considered a part of dialog that allowing a task to be performed. Multi-threading within a interface provides support for multiple tasks to be performed at one time.

Multi-threading can be described as concurrent or interleaved. An interleaved system permits work on a single task at a given time - a word processor allow multiple documents to be open, but only one can be worked on at any instant.

Task migratability

Task migratability means passing responsibility of execution of tasks between user and system. A computerised spell checker is a good example to this. It is a waste of time for a user to manually check a very long document and correct. A spell checking facility in a word processing application can check words against its own computerised dictionary. However, it is considered 'dangerous' to allow a spell-checker program to carry out this task without the user's assistance.

Substitutivity

Substitutivity offers a user alternative ways of specifying input or viewing output. Indeed the distinction between output and input can be blurred. For example, a drawing package may allow start and end co-ordinates of a line to be specified, conversely, the same system may allow the line to be drawn first, and the system indicates the end point co-ordinates.

Customisability

The user interface should be able to support individual preferences. For example standard control bars in MS Word can be amended as required.

Adaptivity can be automated but in order to be able to provide such user-centred system behaviours the system should be trained to distinguish an expert's behaviour from a novice user's behaviour. Repetitive tasks can be detected by observing a user's behaviour and macros can be automatically constructed.

3. Robustness

The robustness of an interface design can be measured in terms of the following four principles. These principles aim to support users to achieve their goals.

Observability

Observability should provide users with an ability to evaluate the internal state from its representation. If a user cannot understand the internal state of the system, there is a high likelihood that the user's confidence will be very low, for example, if the system is performing a time consuming operation, the current status of the operation should be displayed - a web browser will indicate the on-going status of a page download.

There are several aspects to system observability. You should read up more on this topic in your textbook and/or on the Internet.

Recoverability

Users should be able to reach a desired goal after recognition of errors in previous interaction. Error recovery can be achieved in two ways, forward (negotiation) and backward (undo). Forward error recovery involves a user accepting the current state of the system and negotiating from the present state towards the required state. A backward error recovery mechanism within a system allows a user to undo the undesired outcome of the previous interaction by returning to a previous state.

In addition to providing the ability to recover forward or backward, the effort to achieve this should reflect the work being done - the commensurate effort.

Responsiveness

Responsiveness is usually measured in terms of the rate of communication between the system and a user. Response time, indicating change of states within the system, is important. Short duration or instantaneous response time is more desirable.

When an instantaneous response cannot be given by the system, the system should be able to indicate to the user that the system has received the request and in processing an appropriate action (see definition of observability).

As illustrated in the above picture, clicking the print icon on the Microsoft Word menu does not give feedback, e.g. especially novice users may end up pressing the print icon a number of times.