Chapter 6. Design Guides

Table of Contents

Context

In the previous two units we looked at aspects of cognitive psychology and how such knowledge can be used to build more effective computer systems. Researchers and practitioners have formalised the sorts of observation we made into various types of design guides. You will look at the different types of guide and the role they play. The main focus of the unit will be a review and discussion of principles than can be followed to improve a system's' learnability, flexibility and robustness.

Objectives

At the end of this unit you will be able to:

- Distinguish between design standards (e.g. house style and international standards).
- Discuss the role and application of design guides in interactive system development.

- Argue the case for good system learnability, flexibility, and robustness.
- Show how design principles can be used to improve learnability, flexibility, and robustness.

Introduction

The aim of this section is to discuss the role of standardisation in the domain of interface design. Standardisation can be viewed as a way of insuring that good human factors are incorporated in a system. In general, we are familiar with standardisation in many areas of our lives, for example, standard food labels, standard controls on cars, keyboards, colours for traffic lights and so on. The main purpose of standardisation is to provide safer and better quality.

For interactive system designers, designing systems with maximum usability is the ultimate goal. A good design mainly results from knowledge and experience of the designer and more importantly, the way they apply this knowledge into a specific design. In order to increase usability, two kinds of design rules, standards and guidelines, can be applied into a design.

Standards vs Guidelines

The difference between these is that standards are high in authority and limited in application, whereas design guidelines are low in authority and are more general in application.

The best user interface guidelines are high level and contain widely applicable design principles. The designer who intends to apply these principles should know which theoretical evidence supports them and apply the guidelines at an early stage of the design life cycle.

Standards	**Guidelines**
High Authority	Lower Authority
Little overlap	Conflicts, overlap, trade-offs
Limited application - e.g. display area	Less focused
Minimal interpretation-little knowledge required	Interpretation required -expert HCI knowledge

National and international bodies most commonly set interactive system design standards such as British Standards Institution (BSI), the International Organisation for Standardisation (ISO).

Design guidelines can be found in various forms, for example, journal articles, technical reports, general handbooks, and company house style guides. A good example to this is the guidelines produced by Smith and Mosier (1986) . An example of company house style guides is Apple's Human Interface Guidelines. Standardisation in interface design can provide a number of benefits.

Design Guides

Design rules, in the form of standards and guidelines, provide designers with directions for a good and quality design with the intention of increasing the usability of a system.

Software companies produce sets of guidelines for their own developers to follow. These collections are called house standards, sometimes referred to as house style guides. These are more important in industry, especially in large organisations where commonality can be vital. There are two main types of style guides: commercial style guides and corporate style guides.

Hardware and software manufacturers produce commercial style guides and corporate style guides are developed by companies for their own internal use. The advantage of using these guides is to enhance usability of a product through consistency. However, there are other reasons, for example, software developed for Microsoft, Apple Macintosh or IBM PC maintains its "look and feel;" across many product lines.

Though commercial guides often contain high level design principles, in most cases these documents are based on low level design rules which have been developed from high level design principles. If an organisation aims to develop their own corporate style guides for software, generally the starting point will be one of the commercial style guides.

Many design style questions have to be addressed more specifically when developing corporate style guides. For example, how should dialog windows to be organised? Which styles - colour and fonts etc, should be used ?

The roots for design rules may be psychological, cognitive, ergonomic, sociological or computational, but may, or may not be, supported by empirical evidence.

Design Principles

Dix et al (1998) split the design principles into three main categories.

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.

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.

For example, closing a document should always allow the user to save changes not saved already.

Synthesisability

Two aspects of synthesisability are immediate honesty and eventual honesty. In general, these principles relate to the ability of the interactive system to provide the user with an observable and informative notification about the operation state changes within the system.

A good example of this is the file management capabilities of Windows Explorer and the command line operations in DOS. Moving a file from one folder to another is observable by the user in Windows, however, carrying out the same operation in DOS provides no visual representation of the system's actions, in other words no immediate honesty.

The problem with eventual honesty is that if the user is a novice user, not familiar with the system's operations, synthesising the consequences of the operations carried out by the system may be more difficult.

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.

Consistency

To support generalisability, consistency is essential and is probably one of the most widely applied design principle in user interface design. Consistency between application is always favourable, however consistency within an application is essential.

Standard GUI design factors should aid designers to take into account consistency at every level, and, "look and feel" issues should never be abandoned. The use of labels and icons should always be consistent and the same icons and labels should mean the same thing. The principle of "sameness" should be applied to the use of terminology, formatting and input/output behaviour arising from similar situations or task objectives.

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.

A concurrent system allow multiple tasks to be actioned at a given time - within Windows a document can be edited in MS Word and while the file find utility is active.

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.

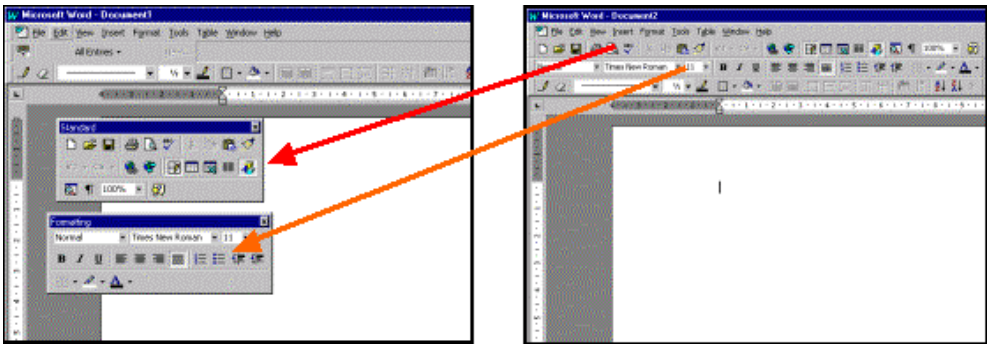
Substitutivity

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.

The customisability principle supports a user's ability to adjust systems settings or features to a form that best suites the preferred way of usage.



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.

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.

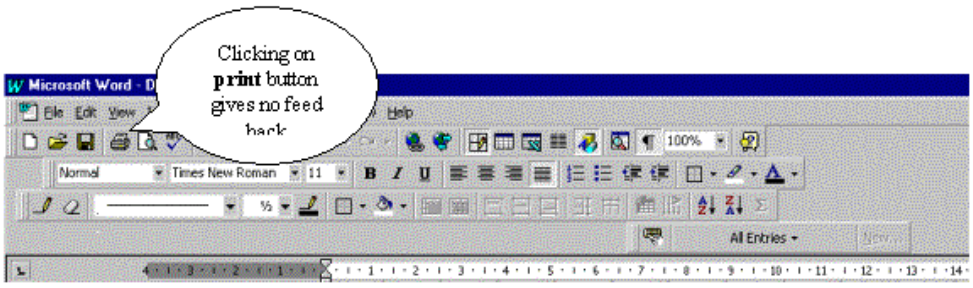
Recoverability

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.

As well as response time of a system, consistency, of responsiveness is also desirable.

Task conformance

There are two aspects of task conformance, task completeness, and task adequacy. Task completeness is concerned with whether a system is capable of supporting the entire task that a user wishes to perform. The task adequacy is concerned with addressing the user's understanding of these tasks It is necessary that an interactive system should allow the user to perform the desired tasks as defined during the task analysis.

Activities & Review Questions

Activity 1 - Consistency for data display interface design

Consider the following menu/actions available within a system.

Case 1	**Case 2**	**Case 3**
delete/insert character	remove/insert character	delete/insert character

Case 1	Case 2	Case 3
delete/insert word	delete/insert word	kill/insert word
delete/insert line	create/insert line	delete/insert line
delete/insert paragraph	kill/create new paragraph	delete/insert paragraph

A good design principle is to create a consistent and familiar user interface. Comment on the above design.

A Discussion on this activity can be found at the end of the chapter.

Activity 2 - Importance of producing standards for interface design

Why is producing standards for interface design important? What kind of benefits can they provide?

A Discussion on this activity can be found at the end of the chapter.

Activity 3 - Design principles and rules

Which of the following can be classified as principles and which of the following can be classified as design rules?

1. Always prompt a 'warning' message to the user before deleting a file.
2. Provide a 'SAVE' command.
3. Reduce cognitive load.
4. Ensure consistency in presentation.
5. Provide a 'HELP' facility.
6. Ensure efficient use.
7. Ensure operational visibility.
8. Display an 'EXIT' command always at the bottom of the 'FILE' menu option .
9. Design for user growth.

A Discussion on this activity can be found at the end of the chapter.

Activity 4 - House style guides - corporate style guides

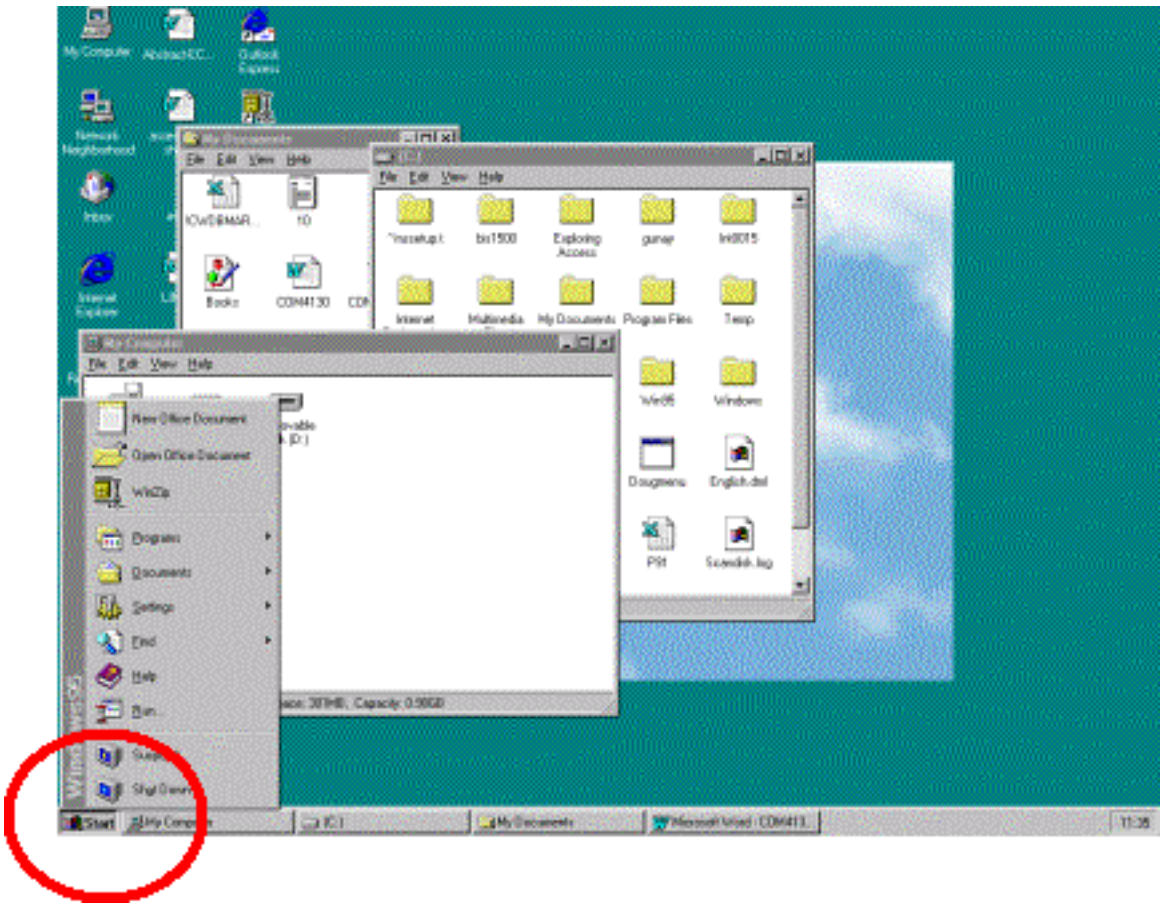
Sketch the screen design of a system to achieve the following. On receiving a house number and postcode from a user, the system must display the location of the address on a map of London. Provide a suitable mechanism to receive input from a user and report errors.

Assume that your design will be implemented using a PC-style interface. Detailed designs of interface items such as buttons and error dialog boxes should be determined at this stage of the design.

A Discussion on this activity can be found at the end of the chapter.

Activity 5 -User interface design principles

Since Windows 95, Microsoft's desktop operating systems' shut down function is accessed via the start menu. Which of the design principles can this be said to contravene?



A Discussion on this activity can be found at the end of the chapter.

Activity 6 - User preemptive dialog

Why is a user preemptive dialog generally preferable? Why is it not always practical?

A Discussion on this activity can be found at the end of the chapter.

Activity 7 - Customisability

What are the advantages of providing a customisable interface? What are the possible dangers?

A Discussion on this activity can be found at the end of the chapter.

Review Questions

1. What advantage could a large software company gain in constraining the creativity of its designers?

   Answer at the end of the chapter.

2. Would you classify the following guidelines as design principles or design rules?

   - Easy to use
   - Adapted to users' level of knowledge
   - System to provide feedback to user

   Answer at the end of the chapter.

3. What problems can an inconsistent interface cause to users?

   Answer at the end of the chapter.

4. What are most design guidelines based on?

   Answer at the end of the chapter.

5. Why is it essential to provide observable feedback when performing an action in an interactive system?

   Answer at the end of the chapter.

6. With reference to task migratability, why may it be dangerous to assign too much control to a system?

   Answer at the end of the chapter.

7. List the three categories of design principles which support the usability of interactive systems.

   Answer at the end of the chapter.

Discussion Topics

Human interface guidelines are often based on experiments. More experimentation could lead to refined standards and defendable. Since we will technology changes very rapidly, we may never have a complete set of guidelines.

Answers and Discussions

Discussion of Activity 1

Although unintuitive the actions for case 1 are consistent. In case 2, there is an inconsistency as an insert operation is present under 'delete' and 'create'. In case 3 a number of characteristics are inherited from case 1, however, access to insert functionality is not consistent. Although no option is ideal, on balance, case 1 would be preferred, case 2 is the least desirable.

A consistent interface is one where similar interactive situations are presented in a similar fashion and exhibits similar behaviours.

Consistency provides a firm foundation for a dialogue to proceed - if a designer ignores the principle of consistency, the user's ability to learn the system and operate effectively will be reduced dramatically.

Discussion of Activity 2

You should read up more on this topic, but the following table may serve as a starting point.

Common terminology	Standard measures of usability or performance mean that designers and user know that they are discussing the same concept.
Maintainability and evolveability	Standard implementation techniques facilitate program maintenance, because all programs can be expected to have a shared style and structure.
Common identity	House or industry standards for display style or screen layout ensure that all systems have the same "look and feel and are easily recognisable.

Less training	Knowledge can be transferred more easily from one system to another if standard command keys and other interaction techniques are adopted by interactive system designers.
Health and safety	Users are less likely to be surprised by unexpected system behaviour if standard controls and warnings are used. Unfriendly systems can be a source of stress.

Discussion of Activity 3

Five of these can be classified as design principles: 3, 4, 6, 7 and 9. The rest 1, 2, 5 and 8 can be considered design rules.

Discussion of Activity 4

Three main elements of style which need to be considered when developing a corporate style guide are:

1. Which style is to be used?

2. How should colour, graphics and other overlays be applied?

3. Which interface components should be used and for which purpose , e.g. when should a button be used, when should a check box be used and so on?

Discussion of Activity 5

Two learnability design principles, familiarity and predictability are contravened. One would not immediately associate the shut down operation with the start button and without prior knowledge one would not have any indication how to shut down the system. However, this is a minor criticism as once the route to the shut down option is learned it is easily accessed again in the future.

Discussion of Activity 6

People have different ways of working A system preemptive approach may enforce a way of working that is not natural for all users. However, a system preemptive approach may be necessary to control certain processes such as a logon procedure before allowing access to an application. It would not be sensible to allow access to a system before verifying logon details. A system-guided approach may also preferred when a task is performed for the first time or for rarely performed tasks

Discussion of Activity 7

New and novice users of an application are likely to initially need support when using a system. As they become more proficient they may develop specialised requirements or find certain features more useful, initial configuration cumbersome.

A customisable interface can be adapted to cater for on going and changing requirements. The danger of customisability is that a system may not be configured adequately for a particular user or a user could be presented with several configurations at different points in time, which may be confusing.

Answer to Review Question 1

First reason is to ensure consistency across all product lines within the organisation and the second reason might be to speed up the design process; if fewer design choices are open to designers, it should take less time to produce a good design.

Answer to Review Question 2

These guidelines are high level principles because they require further interpretation before being applied. (You may wish to refer to Activity 3).

Answer to Review Question 3

Mistakes (some of which may be safety-critical), cognitive overload and, stress.

Answer to Review Question 4

Guidelines may be based on psychological theory, however many are formulated from experience.

Answer to Review Question 5

If there is no observable effect when an action is performed it will not be apparent that anything has happened. If an action cannot be performed a clear explanation as to why should be provided and the status of a long running task should also be displayed.

Answer to Review Question 6

Not withstanding the advances in Artificial Intelligence, computer systems do not always handle situations sensibly. It is thus advisable to offer an option to allow system made changes to be reviewed and deemed appropriate, undone.

Answer to Review Question 7

Learnability, Flexibility and Robustness